MASTER EN SCIENCES INFORMATIQUES

Mise en œuvre d'un langage naturel d'interrogation

Sow, Thierno M.

Award date:
1975

Awarding institution:
Universite de Namur

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

INSTITUT D'INFORMATIQUE

1974-1975

Mise en œuvre d'un langage naturel d'interrogation

Thierno M. SOW

Mémoire présenté pour l'obtention du grade de Maître en Informatique

JURY : H. LEROY

AVANT-PROPOS

Je voudrais exprimer ici toute ma gratitude à :

Mr. H. Leroy - Directeur de ce mémoire, pour les indications précieuses qu'il m'a données sur la conduite de mon travail. Son cours sur la théorie des langages m'a constamment guidé et éclairé.

Mr. W. Paulus - pour ses conseils, ses remarques et sa disponibilité qui m'ont beaucoup encouragé.

Le Père J. Berleur, S.J. - qui m'a aidé à réunir la documentation dont j'avais besoin et qui a bien voulu attirer mon attention sur certains aspects linguistiques du travail.

L'équipe des chercheurs "Grands Fichiers" - pour leurs conseils.

Mme C. Grégoire - qui m'a appris certaines notions de Linguistique et qui a bien voulu relire mon manuscrit

Je remercie les Facultés pour l'aide que j'ai reçue et qui m'a permis de mener à bien ce travail.

INTRODUCTION

De façon générale, on entend par système d'interrogation en langage naturel, un système qui permet à un utilisateur de communiquer avec un ordinateur au moyen de phrases choisies du Langage ordinaire. L'ensemble de ces phrases forme le langage naturel d'interrogation. Les systèmes d'interrogation sont logiquement structurés en trois composantes :

- une composante d'analyse syntaxique qui permet de dégager la structure syntaxique des phrases
- une composante d'analyse sémantique qui interprête cette structure syntaxique en termes de commandes
- une composante de recherche (ou de mise à jour) qui réalise ces commances.

Dans ce travail, nous nous proposons de définir les deux premières composantes d'un système d'interrogation qui accepte un langage naturel d'interrogation aussi proche que possible du français usuel.

Nous supposons que ce langage naturel, qui est nécessairement un sous-ensemble réduit du langage ordinaire, porte sur une base de données B, qui à titre d'exemple, est un catalogue d'une salle de lecture de Bibliothèque.

Dans un système d'interrogation, le but de l'analyse syntaxique est non pas d'établir la "grammaticalité" des phrases mais plutôt d'associer à chaque phrase une structure syntaxique qui en permette une interprétation non ambiguë. Le premier problème qui se pose alors est celui de la définition d'un modèle syntaxique adéquat :

- ce modèle doit permettre de déterminer correctement la structure syntaxique des phrases.

- ce modèle doit caractériser certaines subtilités des langages naturels comme les relations qui existent entre les formes affirmative, négative et interrogative d'une même phrase.

- ce modèle doit conduire à des algorithmes de reconnaissance simples et efficaces.

La théorie des grammaires transformationnelles de Chomsky résout en partie ces problèmes. Mais outre le fait que ces modèles sont totalement orientés vers la génération des phrases et non leur analyse (problème qui nous occupe ici), ces modèles n'ont pas permis de construire des analyseurs efficaces du langage naturel (Woods [36]).

Nous avons dont utilisé une approche catégorielle dont l'intérêt essentiel est de pouvoir définir un calcul syntaxique de la structure d'une phrase. Si dans une grammaire context-free, on considère par exemple qu'un article suivi d'un nom forme un syntagme nominal, dans une grammaire catégorielle, l'article est considéré comme un opérateur qui transforme le nom en nom ÷ la combinaison des constituants en syntagmes est considérée comme le résultat d'une opération de l'un des constituants sur les autres. Ainsi on se donne un ensemble de catégories grammaticales munies chacune d'indicateurs qui précisent de façon univoque les opérations qu'elles peuvent faire subir aux catégories qui les entourent dans une suite donnée. Ces opérations dites règles de réduction sont analogues aux règles arithmétiques de multiplication et de division, et définissent le calcul syntaxique.

Mais dans la grammaire de Bar-Hillel (chapitre 2) ce calcul n'est pas associatif ce qui signifie en pratique qu'il

faut faire plusieurs lectures avant de trouver les bonnes réductions.

Nous définissons (chapitre 5) une grammaire catégorielle G_1 basée sur une règle additive qui évite cet inconvénient. Une grammaire catégorielle étant fondamentalement équivalente à une grammaire context-free il ne peut être question d'utiliser G_1 pour reconnaître tout le langage naturel (chapitre 6). G_1 permettra de construire une structure partielle de la phrase et un réseau grammatical élargi fera le reste de la reconnaissance.

Le réseau grammatical élargi (Woods [36]) offre les mêmes possibilités qu'un automate à pile non déterministe et permet de réaliser l'équivalent d'une reconnaissance transformationnelle sans utiliser une composante transformationnelle. Par un choix judicieux de conditions et d'actions, le principe est de mémoriser dans des registres, les informations sur la structure profonde de la phrase tout en en dégageant la structure de surface (chapitre 2).

L'analyseur syntaxique construit sur ces principes, réalise l'analyse syntaxique de 15 phrases de longueur variable (inférieure à 20 mots) et de types différents (affirmative, interrogative, impérative) en moins de 4 s. de temps C.P.U. (chapitre 10).

Pour l'interprétation sémantique, nous avons repris et adapté au français le modèle sémantique de Bobrow tel qu'il a été développé par Woods dans sa thèse (Woods [34]) avec essentiellement l'introduction de quantificateurs.

Nous avons donc défini (chapitre 7) un modèle sémantique pour la base de données B et un langage de commande qui constitue un interface entre l'interprétation sémantique et la composante de recherche, libérant ainsi le système d'interrogation d'une structure particulière des données.

L'interpreteur sémantique travaille sur un ensemble de règles sémantiques dont la composition peut être modifiée sans que le fonctionnement de l'interpréteur en soit affecté.

L'interprétation d'une phrase du langage naturel se ramène à :

- la détermination des objets désignés par les syntagmes nominaux de la phrase (règles sémantiques relatives aux noms)

- la détermination des relations qui unissent ces objets; ces relations sont exprimées par des verbes suivis éventuellement de prépositions (règles sémantiques relatives aux verbes).

Ce qui nous amène à distinguer (Woods) un interpréteur des syntagmes nominaux (NP-Processor) et un interpréteur des syntagmes verbaux (S-Processor) - qui traduisent la phrase en une expression du langage de commande qui est considéré comme l'interprétation sémantique de la phrase.

Le mémoire s'articule de la façon suivante :
dans la première partie, nous présentons les concepts fondamentaux et nous essayons de justifier les choix qui vont nous permettre en deuxième partie de proposer un langage naturel et le système d'interrogation qui s'appuie sur ce langage. La troisième partie présente l'implémentation de l'analyseur syntaxique.

TABLE DES MATIERES

I. CONCEPTS FONDAMENTAUX

Nous avons retenu dans cette partie les notations utilisées (et communément admises) des différents auteurs. Les catégories grammaticales sont notées comme suit selon que nous donnons des exemples en anglais ou en français:

En anglais	En Français	Catégorie grammaticale
S	S	phrase
NP	SN	syntagme nominal
VP	SPRD	syntagme prédicatif
V	SVER	syntagme verbal
	V, VER	verbe
AUX	AUX	auxiliaire
TPS	TPS, TNS	temps
PRES	PRES	présent
PP	SPP	syntagme prépositionnel
N	N, NCO	nom commun
NPR	NPR	nom propre
PRO	PRO	pronom
NU	NU	nombre
SG	SG	singulier
PL	PL	pluriel
DET	DET	déterminant
NEG	NEG	signe de négation
ADJ	ADJ	adjectif
ADV	ADV	adverbe
PREP	PREP	préposition
Q	Q	signe d'interrogation

1. LES SYSTEMES D'INTERROGATION

1.1. Définition

Supposons définie une base de données B. Un système d'interrogation est un système qui, partant d'une phrase exprimée dans un langage naturel, permet :

- soit d'obtenir des informations sur des éléments de B
- soit de faire une mise-à-jour de B

De façon générale, on distingue trois composantes dans ce système :

1) Une composante dite d'analyse syntaxique qui reçoit en entrée la phrase libellée en langage naturel et en dégage la structure syntaxique - c'est-à-dire les relations grammaticales entre les différents constituants de la phrase. Cette structure syntaxique est ordinairement présentée sous la forme d'un arbre syntaxique
2) Sur la base de cette information, la composante dite d'analyse semantique construit une représentation formelle du contenu semantique ou interprétation sémantique de la phrase.
3) La troisième composante dite de recherche met en oeuvre des procédures choisies en fonction de l'interprétation sémantique de la phrase pour selon le cas, répondre à une question posée sur des éléments de B ou mettre à jour B.

La structure logique d'un système d'interrogation est donc de la forme:

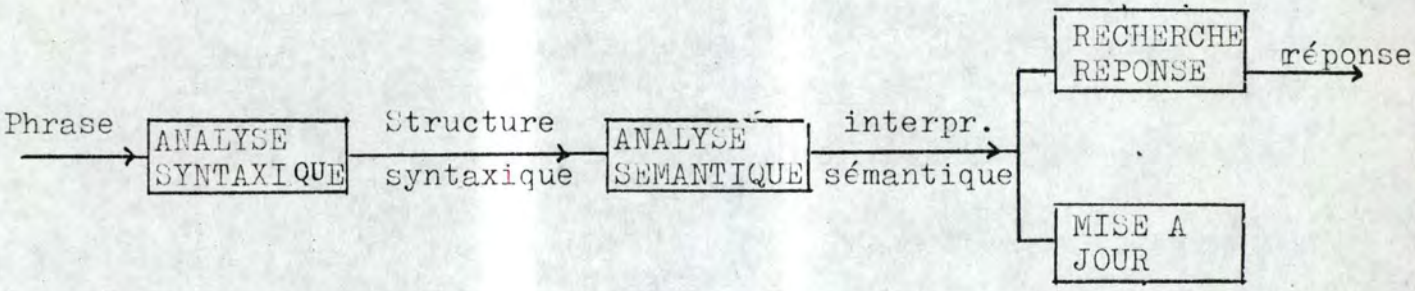


Figure 1

Cependant l'analyse sémantique n'est pas toujours distincte de l'analyse syntaxique et de plus, dans certains systèmes, ces trois composantes sont réalisées en parallèle. Dans ce cas, il y a un échange d'informations entre elles symbolisé par les traits discontinus.

1.2. Historique

On distingue (Simmons [26, 28]) deux générations de systèmes d'interrogation :

La première (1959-1965) a servi essentiellement à poser les principaux problèmes liés à l'analyse et l'interprétation automatique des langages naturels. Ces systèmes ont été handicapés pour deux raisons : l'absence de modèle linguistique adéquat et le fait que les langages de programmation n'étaient pas bien adaptés à l'implémentation.

La deuxième génération (après 1965) correspond à l'apparition de nouvelles théories linguistiques - principalement les grammaires transformationnelles. On construit aussi de nouveaux langages évolués comme LISP et ALGOL.

Les différents systèmes peuvent être rangés en quatre groupes :

a) <u>systèmes conversationnels</u> (conversation machines). On leur prête une certaine intelligence qui leur permet de tenir une conversation sur un sujet donné.

Exemple de conversation que permet ELIZA [32] (les réponses du système sont en majuscules) :

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE ?

He says I'm depressed much of the time

I AM SORY TO HEAR YOU ARE DEPRESSED etc...

A l'entrée d'une phrase comme :

You are very helpful

Le mot "you" est sélectionné comme mot-clé. Un dictionnaire permet de faire correspondre à chaque mot-clé différents modèles qui peuvent s'appliquer. Dans ce cas-ci le modèle :

(o you are o)

qui ignore tout ce qui précède et tout ce qui suit l'expression "you are", convient. A ce modèle est associé une opération :

(what makes you think I am 4)

dans laquelle il faut remplacer le 4 par le 4è constituant de la phrase d'entrée.

Cette phrase est décomposée comme suit :

1 : (empty); 2 : you; 3 : are; 4 : very helpful;

ELIZA remplace alors 4 par "very helpful" et donne la réponse suivante :

WHAT MAKES YOU THINK I AM VERY HELPFUL ?

b) <u>Résolution de problèmes mathématiques</u> (Mathematical word - problem processors). Le principe de ces systèmes est de réduire les phrases en équations algébriques qu'ils sont en mesure de résoudre.

Exemple tiré de STUDENT (système de résolution de problèmes arithmétiques construit par Bobrow). Soit la phrase :

If the number of customers TOM gets is twice the square of 20 percent of the numbers of advertisements he runs, and the number of advertisements he runs is 45, what is the number of customers TOM gets ?

L'analyse se fait en trois phases :

Une première phase permet de faire des remplacements : "twice" par "2 times", the"square of" par "square". La deuxième phrase consiste à l'identification d'opérateurs ("plus", "percent", "times", "of"...), de certains verbes (VERB), des mots interrogatifs (AWORD) et du point d'interrogation (WMARK) qui termine la phrase (DLM). Après.ces opérations, on obtient une expression de la forme :

If the number (of/OP) customers TOM (gets/VERB) is 2 (times/OP1) the square/OP1) 20 (percent/OP2) (of/OP) the number (of/OP) advertisements (he/PRO) runs, and the number (of/OP) advertisements (he/PRO) runs is 45 (what/QWORD) is the number (of/OP) customesrs TOM (gets/VERB) (WMARK/DLM).

La dernière phrase consiste à la séparation de cette phrase en phrases plus simples par l'analyse des connecteurs (` , ' , ` et,.) :

1.- the number of customers TOM gets is 2 times the square 20 pescent of the number of advertisement he runs
2.- the number of advertisements he runs is 45
3.- what is the number of customers TOM gets QMARK.

Ces phrases de la forme "—— is ——" donnent alors les équations cherchées.

c) Analyse de textes (Natural language text processing). Ces systèmes s'apparentent à des modèles psychologiques et tentent d'analyser le langage naturel en termes de structures qui représentent ce qu'il y a de typique dans les différents concepts du langage.

Par exemple, dans le TLC (Teachable Language Comprehender de Quillian [28]) les différentes relations syntaxiques et sémantiques entre concepts sont représentées par un réseau. A la lecture de l'expression :

"Client's lawyer"

le système enregistre les deux faits suivants :

(1) "client" est une personne qui emploie un professionnel

(2) "lawyer" est un professionnel qui représente une personne devant la loi.

En comparant "client" et "lawyer" le système découvre le point d'intersection "professionnel" qui est d'une part une propriété de "client" de l'autre, la classe de "lawyer". Cette intersection lui permet d'établir la relation : "lawyer" est un professionnel employé par "client".

d) systèmes de déduction (Fact. retrieval Systems).

Le but de ces systèmes est la recherche d'une information qu'il faut déduire :

- soit à partir d'une base de données : c'est par exemple le système de Woods qui sera introduit plus loin.

- soit à partir de règles de déduction et d'informations communiquées au système.

Exemple : le DEDUCOM (Deductive Communicator). On communique au système les données suivantes :

1. une main a 5 doigts
2. un bras a une main
3. un homme a 2 bras

et la règle de déduction suivante :

4. Si V a m objet X et
   si Y a n objets V alors
   Y a n m objets X

En substituant les données dans la règle de déduction, DEDUCOM répond à la question suivante :

Question : Combien de doigts a un homme ?

Réponse : 10

1.3. Remarques.

Woods [34] souligne que dans la majorité des cas, les méthodes de traitement utilisées dans les trois composantes d'un système d'interrogation sont largement influencées par la structure des données mémorisées dans l'ordinateur.

Dans des cas extrêmes, l'analyse syntaxique est faite avec une grammaire dont les éléments ne correspondent pas à des catégories grammaticales mais à des structures de données. Et le système devient caduc dès que ces structures changent.

De même dans beaucoup de systèmes, l'organisation des données sert de "modèle sémantique". De sorte que l'interprétation sémantique se fait plus ou moins, en termes de structures de données. L'interpreteur devient ainsi complétement dépendant de la structure actuelle des données.

Nous allons maintenant examiner le problème du choix d'un modèle syntaxique ou grammaire et de la définition d'un modèle sémantique qui libèrent le système d'interrogation de l'organisation des données.

2. MODELES SYNTAXIQUES

Dans ce paragraphe nous examinons le problème du choix d'une grammaire qui permette d'analyser les phrases que le système reçoit en entrée.
Cette question touche à la linguistique, mais le problème n'est examiné ici que du point de vue traitement automatique par exemple: difficultés d'implémentation, encombrement mémoire, temps d'analyse, etc...

Nous avons utilisé les références suivantes : Woods [34,36] Chomsky [9] , Ruwet [22] , J. Friedman [15] .

Pour poser une question relative à un ensemble d'éléments de la base de donnée B, on utilise normalement des phrases interrogatives ou impératives dont le but est de déclencher une procédure de recherche de l'information voulue :

. Quelle est la composition du produit A ?

. Donnez la composition du produit A.

. Les phrases déclaratives peuvent servir à mémoriser une certaine information dans B :

A est un nouveau produit

Le produit A est disponible

Le produit A n'est pas disponible.

qui disent respectivement qu'il faut créer un nouvel objet A dans la catégorie des produits; il faut marquer que le produit A est disponible; il faut marquer que le produit A n'est pas disponible.

Toutes ces phrases sont destinées à faire réaliser à la

composante de recherche une action bien précise en relation avec B (recherche d'information ou mise-à-jour).
Nous les appellerons des commandes. Dans tout ce qui suit nous n'avons en vue que ce type de phrases.

Considérons la phrase :

Alpha fabrique le produit A (1)

et les questions suivantes :

Alpha fabrique-t-il le produit A ? (2)

Que fabrique Alpha ? (3)

Qui fabrique le produit A ? (4)

(2) appelle la réponse "oui" ou "non" selon que (1) est vraie ou pas c'est-à-dire si oui ou non la proposition :

X fabrique Y

est vraie, lorsqu'on remplace X par "Alpha" et Y par "le produit A".

La réponse à (3) est l'ensemble des Y, qui rendent (5) vraie pour X remplacé par "Alpha".

De même la réponse à (4) est l'ensemble des X qui rendent (5) vraie pour Y remplacé par "le produit A".

Cela signifie que bien que la nature des réponses à (2) d'une part et (3)-(4) de l'autre soit très différente : oui/non dans un cas, un ensemble d'objets dans l'autre, répondre à ces trois questions revient à déterminer la véracité de (5) pour différents valeurs de X et Y.

L'interprétation des phrases interrogatives se ramène donc à l'interprétation des phrases déclaratives.
Ce sont donc ces dernières qu'il faut savoir analyser et interpréter en priorité. Un autre problème sera celui de ramener une phrase interrogative sous sa forme affirmative.

2.1. Grammaires Transformationnelles

Le premier modèle proposé par Chomsky est celui d'une grammaire CF (context-free). Mais une telle grammaire ne peut caractériser toutes les subtilités d'un langage formel (par exemple le problème de la liaison entre un identificateur et sa déclaration) et à fortiori celles d'une langue naturelle. Par exemple, une grammaire CF ne permet pas de montrer les relations qui existent entre la forme déclarative d'une phrase et sa forme interrogative ou bien entre sa forme active et sa forme passive.

La théorie de Chomsky d'une grammaire transformationnelle dans laquelle on distingue la structure de surface d'une phrase et sa structure profonde qui porte l'interprétation sémantique de la phrase, résout en partie ce problème.

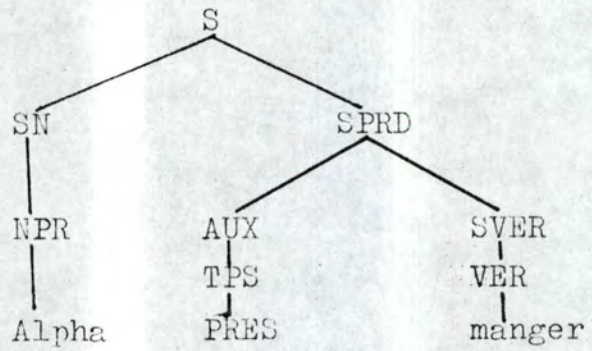
Considérons par exemple les trois phrases :

Alpha mange (5)

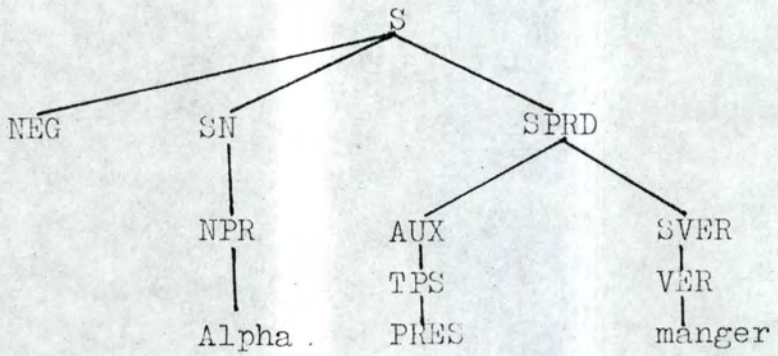
Alpha ne mange pas (6)

Alpha mange-t-il ? (7)

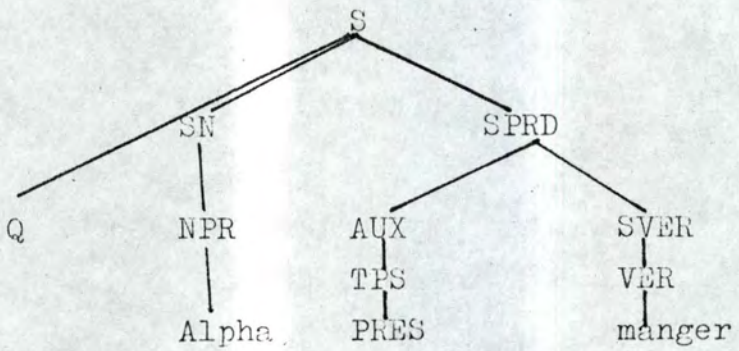
Du fait que, comme nous l'avons montré ci-dessus, (6) et (7) se ramènent à l'évaluation de la fonction de vérité de la proposition (5); on associe à ces trois phrases la même structure profonde complétée de deux noeuds supplémentaires pour indiquer respectivement la forme négative de (6) et la forme interrogative de (7). Ainsi les structures profondes seront représentées respectivement de la façon suivante :



(5')



(6')

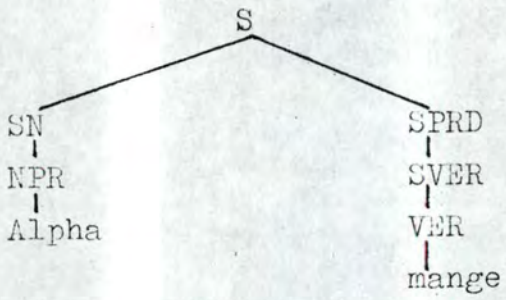


(7')

où la branche AUX-TPS-PRES indique l'auxiliaire du temps : le présent de l'indicatif.

On se donnera ensuite un ensemble de transformations qui permettent de dégager de (5'), (6') et (7') les structures de surface correspondantes.

Ainsi, pour (5') une transformation dite d'accord [22] permettra de faire la conjugaison de manger au présent et son accord au sujet Alpha pour donner :



(5")

qui est la structure de surface de (5)

De même pour (6') et (7') on définira en plus de cette transformation d'accord, deux autres transformations :

- T_{NEG} qui appliquée à (6') va supprimer le noeud NEG et générer l'adverbe de négation

ne ——— pas

dans le noeud SPRD [22]/

- T_Q qui appliquée à (7') va supprimer le noeud Q et générer le pronom 'il' et éventuellement le 't' de liaison dans SPRD [18].

La distinction entre structure de surface et structure profonde permet en outre de résoudre certaines situations ambiguës. Soit par exemple les deux phrases :

le professeur est prêt à partir (8)

le professeur est difficile à contenter (9)

Ces phrases ont la même structure de surface. Mais on constate que le syntagme nominal 'le professeur' ne joue pas le même rôle vis-à-vis de l'infinitif. Dans (8) 'le professeur' est sujet de 'partir' alors que dans (9)

'le professeur' est objet de 'contenter'.

Ainsi en structure profonde on fera ressortir pour (8) les deux phrases sous-jacentes :

le professeur est prêt (8')
le professeur part (8")

et pour (9) :

le professeur est difficile (9')
quelqu'un contente le professeur (9")

qui permettront des interprétations non ambiguës de (9) et (10)

Une grammaire transformationnelle comprend deux composantes

- une grammaire CF dite composante de base
- un ensemble de règles transformationnelles dit composante transformationnelle.

2.1.1. La composante de base

Le but de cette composante est de construire la structure profonde de la phrase. Mais l'analyse qui nous occupe ici vise non pas à étudier la "Grammaticalité de la phrase" mais à lui assigner une structure qui permette une interprétation non ambiguë de la phrase.

Ainsi, il n'y aurait aucune raison de rejetter des phrases considérées comme agrammaticales si le système est capable de les interpréter. De même des détails grammaticaux comme le genre, le nombre, les accords, ... ne sont importants que s'ils résolvent une situation qui dans le cas contraire serait ambiguë.

2.1.2. La composante transformationnelle

Elle contient les règles transformationnelles, qui appliquées à la structure profonde permettent de dégager la structure de surface de la phrase. Ce processus se réalise au moyen d'un certain nombre d'opérations sur des parties de la structure profonde : déplacements, recopies et effacements.

Dans cette composante, sont spécifiés :
- l'ordre dans lequel les transformations sont appliquées
- et pour chaque transformation T :

a) son domaine, c'est-à-dire la ou les structures auxquelles T s'applique

b) le changement structural qui est l'ensemble des opérations que T fait subir à son domaine (substitution, suppression, addition, ...).

c) la structure qui doit résulter de ce changement.

En pratique, un système basé sur l'utilisation d'une grammaire transformationnelle s'alourdit très vite du fait que ces transformations agissent sur des arbres syntaxiques qu'il faut d'abord décrire ensuite generer, reconnaître ou modifier selon le cas.

Exemple : J. Friedmann [15] donne les indications suivantes sur l'implémentation d'un modèle de grammaire transformationnelle en FORTRAN IV : 10.000 lignes (y compris les commentaires) et un code objet de 300.000 octets environ sur IBM 360/67.

2.2. Grammaires élargies (augmented grammars)

Les grammaires transformationnelles telles qu'elles sont définies, sont totalement orientées vers la génération

des phrases et non leur analyse - problème qui nous occupe ici. Woods [34, 36] cite et critique de la façon suivante quelques tentatives qui ont été faites d'utilisation de ces grammaires pour analyser des phrases.

Matthews propose un"algorithme d'analyse par synthèse" : il s'agit d'appliquer les règles transformationnelles pour générer toutes les phrases possibles tout en regardant si une des phrases obtenues se ramène à la phrase qu'on analyse. Mais cet algorithme est si inefficace qu'il n'est d'aucune application pratique.

Deux autres méthodes (Mitre, Petrick) sont basées sur l'utilisation de transformations dites "inverses" (inverse transformation) pour inverser le processus de génération : partir de la structure de surface pour obtenir la structure profonde.

Cependant il n'existe pas dans le modèle transformationnel une composante pour décrire explicitement la structure de surface. Dans les deux cas, on résout ce problème en construisant une grammaire dite grammaire élargie (augmented grammar) qui comprend outre les règles de la grammaire CF de base quelques règles additionnelles qui caractérisent les structures qui peuvent résulter des transformations.

Dans le cas du Mitre, cette grammaire est construite à la main : il n'y a aucune procédure formelle qui permette de la déduire de la grammaire transformationnelle originelle. La grammaire élargie est utilisée pour construire une structure de surface complète sur laquelle opèrent par la suite les "transformations inverses".

Dans le système de Petrick, la grammaire élargie est utilisée pour construire une structure partielle sur laquelle sont appliquées les "transformations inverses". Dans les deux systèmes, les "transformations inverses" peuvent donner des structures profondes qui ne sont pas valides.

Les inconvénients de ces méthodes tiennent à la difficulté de déterminer les "transformations inverses" et aussi au temps élevé d'analyse qu'elles demandent (de l'ordre d'une heure dans la première version de l'algorithme de Petrick pour reconnaître une phrase simple - sur UNIVAK M-460).

2.3. Réseau grammatical élargi

(Augmented Transition Network grammar)

2.3.1. Réseau grammatical

L'idée de base d'un réseau grammatical est la fusion des parties droites des règles d'une grammaire CF qui ont la même partie gauche, en un seul diagramme de transition.

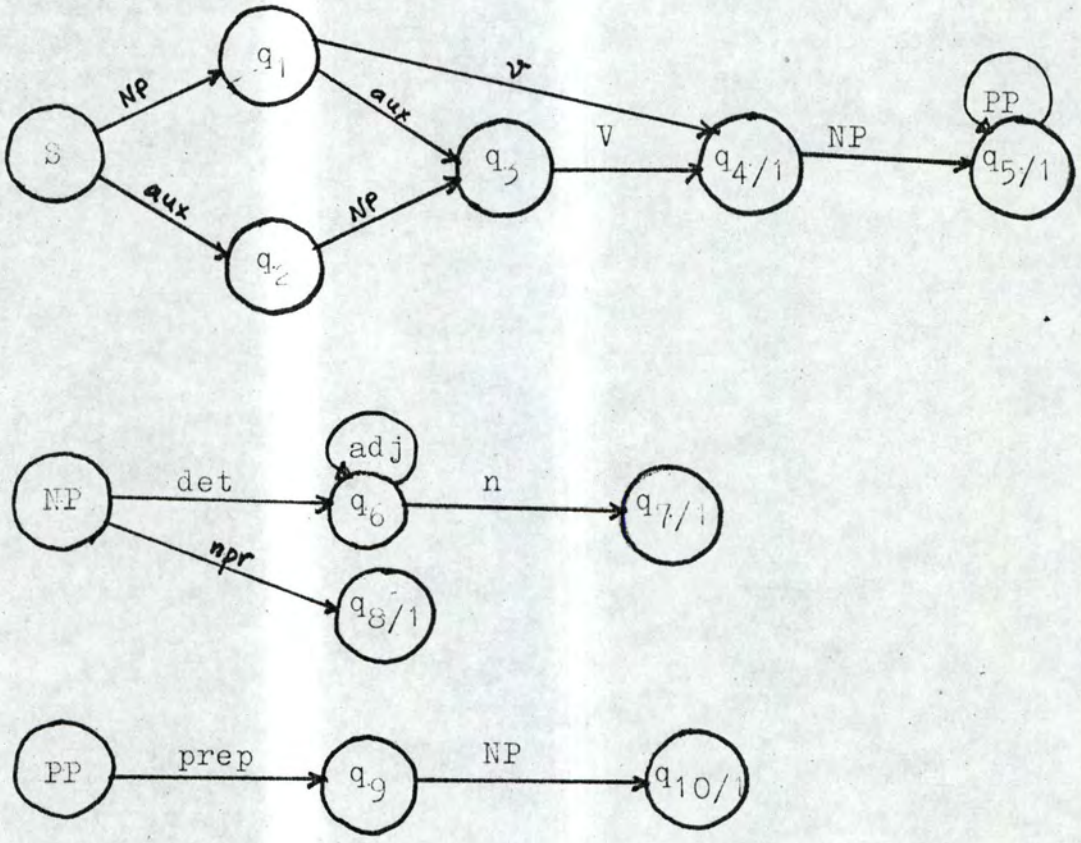
Un réseau grammatical est un graphe orienté avec des arcs et des états référencés, un état distingué dit état initial et un ensemble distingué d'états dits états acceptants. Les labels indiqués peuvent être aussi bien des noms d'états que des symboles terminaux.

L'interprétation d'un arc référencé par un nom d'état E est que l'état F du réseau à la fin de l'arc sera sauvé sur une pile et le contrôle passera (sans lecture de nouveau symbole) à E. C'est-à-dire qu'on permet la transition vers l'état E dès qu'on a reconnu dans la chaîne à analyser une construction du type E.

Quand un état terminal est rencontré, la pile est "épurée" en enlevant l'état E_s qui se trouve au sommet de la pile et en transférant le contrôle à E_s.

La chaîne est acceptée si la pile est vide après analyse du dernier élément de la chaîne.

Exemple



où S est à l'état initial

q_4, q_5, q_7, q_8, q_{10} - les états acceptants soulignés d'un 1.

Le fonctionnement du réseau sera expliqué plus loin. Le réseau grammatical est en fait équivalent à un automate à pile non déterministe.

2.3.2. Réseau grammatical élargi

En 1965, Kuno (cité par Woods [36]) émet l'idée qu'il devrait être possible d'améliorer une grammaire élargie de façon qu'elle mémorise d'une certaine façon l'équivalent de la structure profonde d'une phrase en construisant la structure de surface et ce sans recourir à une composante de "transformations inverses".

Woods [36], à la suite d'autres auteurs, utilise cette idée pour construire un réseau grammatical élargi qui est un réseau grammatical doué d'un mécanisme qui lui permet de réaliser l'équivalent d'une analyse transformationnelle sans composante séparée de"transformations inverses".

Nous reprenons ici un fragment de ce réseau pour illustrer son fonctionnement.

Soit le réseau suivant (exprimé en LISP) :

```
(S / (PUSH NP/T (SETR SUBJ *) SETR TYPE (QUOTE DCL)) (TO Q1))
     (CAT AUX T (SETR AUX *) SETR TYPE (QUOTE Q)) (TO Q2))

(Q1 (CAT V T (SETR AUX NIL) SETR V *) TO Q4))
    (CAT AUX T (SETR AUX *) TO Q3)))

(Q2 (PUSH NP/T (SETR SUBJ *) (TO Q3)))
(Q3 (CAT V T (SETR V *) (TO Q4)))
(Q4 (POP (BUILDQ (S +++(VP +)) TYPE SUBJ AUX V) T)
    (PUSH NP/T (SETR VP (BUILDQ (VP (V +) *) V)) (TO Q5)))

(Q5 (POP (BUILDQ (S ++++) TYPE SUBJ AUX VP) T)
    (PUSH PP/T (SETR VP (APPEND (GETR VP)(LIST *)))(TO Q5)))
```

qui est la portion élargie qui correspond à la première partie du réseau donné précédemment (états S, q_1, q_2, q_3, q_4, q_5).

Reprenons la phrase donnée en exemple par Woods :

Does John like Mary?

1.- On commence l'analyse à l'état initial S/ en lisant le premier mot "does" qui est auxiliaire : AUX. L'arc aux (CAT AUX T...) peut donc être suivi :
- (SETR AUX *) initialise le registre AUX avec "does"
- (SETR TYPE (QUOTE Q)) initialise le registre TYPE avec le symbole Q
- (TO Q2) fait passer le réseau à l'état Q2 avec lecture du mot suivant : "John".

2.- Il y a un seul arc qui part de Q2. Cet arc étant referencé par un nom d'état NP, on empile Q3 et le réseau passe à l'état NP qui va retourner la valeur (NP John).
NP étant reconnu, on passe à la réalisation des actions spécifiées dans l'arc :

(SETR SUBJ *) met la valeur "(NP John)" dans le registre SUBJ et (TO Q3) ramène le réseau à Q3 (qui est ainsi enlevé du sommet de la pile) avec lecture du mot suivant : "like".

L'état actuel des registres est :

TYPE : Q
AUX : does
SUBJ : (NP John)

3.- Le verbe "like" fait passer le réseau de Q3 à Q4 après lecture du mot suivant "mary" après avoir mis la

valeur "like" dans le registre V.

4.- Q_4 est un état acceptant. L'arc "POP" qui indique que toute la chaîne est analysée et qu'elle est une phrase ne peut être suivi du moment que toute la chaîne n'est pas traitée. On suit donc l'arc

(PUSH NP/ ...)

qui fait empiler Q5 et passe le contrôle à l'état NP qui retourne la valeur "NP Mary".

SETR VP (BUILDQ (VP (V +) *) V)) va remplacer dans la structure (VP (V +) *), * par "NP Mary" et + par le contenu du registre indiqué V. D'où la structure résultante :

(VP (V like) (NP Mary))

qui va être placée dans le registre VP.

(TO Q5) ramène le réseau à l'état Q5 qui est enlevé du sommet de la pile (qui est maintenant vide).

L'état actuel des registres est le suivant :
TYPE : Q
AUX : does
SUBJ : (NP John)
V : like
VP : (VP (V like) (NP Mary))

5.- On est à la fin de la phrase et comme Q5 est un état acceptant (c'est-à-dire qu'il possède un arc "POP") et que la condition T est satisfaite , la chaîne est acceptée comme une phrase.

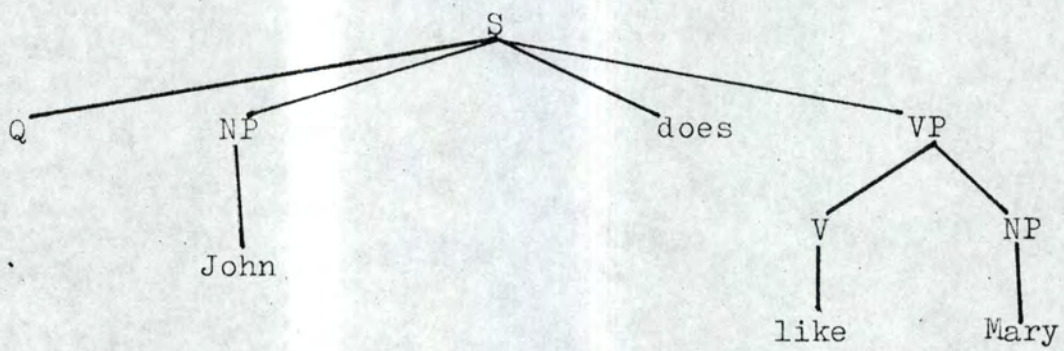
La forme :

(BUILDQ (S ++++) TYPE SUBJ AUX VP)

indique la valeur à retrouver comme analyse de la phrase. Cette valeur est obtenue en remplaçant les plus "+" successifs respectivement par TYPE, SUBJ, AUX et VP, d'où la structure finale :

(S Q (NP John) does (VP (V like) (NP Mary)))

qui représente l'arbre syntaxique suivant :



2.4. Grammaire catégorielle

La théorie des grammaires catégorielles telle qu'elle est exposée ici a été essentiellement développée par Bar-Hillel. Les références utilisées sont Bar-Hillel [1, 2, 4], Marcus [19], Bobrow [6] et Chomsky [9].

2.4.1. Définition

Une grammaire catégorielle G_o est un quintuplet :

$$G_o = (V, C, S, R, f) \qquad \text{où}$$

1) V est un ensemble fini appelé vocabulaire
2) C est un ensemble fini d'éléments dits catégories fondamentales

$$C_1, C_2, \ldots, C_4$$

C est fermé pour les opérations de réduction gauche et droite notées respectivement $\backslash$ et $/$:

$$C_1 \in C \qquad (C_1 \backslash C_2) \in C$$
$$C_2 \in C \qquad (C_1 / C_2) \in C$$

$(C_1 \backslash C_2)$ et (C_1 / C_2) sont dites catégories dérivées.

3) S est la catégorie distinguée
4) R est un ensemble de deux règles de réduction :

$$R_1 : \quad (C_i / C_j)\ C_j \longrightarrow C_i \qquad C_i \in C,\ C_j \in C$$
$$R_2 : \quad C_i\ (C_i \backslash C_j) \longrightarrow C_j$$

qui signifient respectivement que :
- la catégorie (C_i / C_j) suivie de C_j se réécrit C_i
- la catégorie C_i suivie de $(C_i \backslash C_j)$ se réécrit C_j

5) f est une application de V dans un sous-ensemble de C, appelée fonction d'assignation.

On notera que les règles R sont analogues aux règles arithmétiques de multiplication et de division :

$$\frac{C_i}{C_j} * C_j = C_i$$
$$C_i * \frac{C_j}{C_i} = C_j$$

Soit α, β deux chaînes de catégories :
on dit que α se réduit directement à β

$$\alpha \Longrightarrow \beta$$

Si β résulte de α par l'application de l'une des règles R.
On dit que α se réduit à β

$$\alpha \overset{*}{\Longrightarrow} \beta$$

s'il existe une suite de catégories $C_1, \ldots C_p$ telle que :

$$\alpha = C_1, \quad \beta = C_p \text{ et } C_i \Longrightarrow C_{i+1} \qquad i = 1, \ldots, p-1$$

Une chaîne x, d'éléments de V est une phrase si au moins une des catégories associées à x par f se réduit à S.
L'ensemble de toutes les phrases forme le langage engendré par G_o.

Exemple : soit V = {Alpha, dort, construit, maison, très, une, grande}, C = {S, n}. n est la catégorie dite nom, S la catégorie phrase.
Définissons f de la façon suivante :

f(Alpha) = f(maison) = n
f(dort) = (n \ S)
f(construit) = ((n \ S)/n)
f(une) = f(grande) = (n / n)
f(très) = ((n / n) / (n / n))

Ainsi le verbe intransitif 'dort' est considéré comme un opérateur qui transforme le nom qui apparaît à sa gauche en phrase.

Le verbe transitif 'construit' est un opérateur qui transforme le nom apparaissant à sa droite en verbe intransitif.

L'article 'une' et l'adjectif 'grande' transforment les noms qui apparaissent à leur droite en nom.

L'adverbe 'très' transforme un adjectif apparaissant à sa droite en adjectif.

Ces réductions se font de la façon suivante :

$$\frac{\text{Alpha}_{n} \qquad \text{dort}_{n\backslash S}}{S} \qquad (1)$$

$$\frac{\dfrac{\dfrac{\text{Alpha}_{n}\ \text{construit}_{(n\backslash S)/n}\ \text{une}_{n/n}\ \text{très}_{(n/n)/(n/n)}\ \text{grande}_{n/n}\ \text{maison}_{n}}{n/n}}{n}}{n}}{n\backslash S}}{S} \qquad (2)$$

Une réduction est une succession de lignes dans laquelle chaque ligne diffère de la précédente par le fait que deux catégories adjacentes ont été combinées pour former une nouvelle catégorie.

2.4.2. Propriétés de réductions

1.- Les réductions ne sont pas commutatives.
En effet, si dans (1) on permute Alpha et dort on obtient la chaîne : n $\backslash$ S n et aucune des deux règles de réduction ne lui est applicable.

2.- Les réductions ne sont pas associatives.
La réduction (2) peut se noter :

$$\left[n \left[\left(\left(n\backslash S\right)/n\right) \left[n/n \left[\left[\left(\left(n/n\right)/\left(n/n\right)\right) n/n\right] n\right]\right]\right]\right] \qquad (2')$$

La seule autre tentative possible aurait été de réduire grande et maison :

$$(n/n)n \longrightarrow n$$

On aurait obtenu

$$n \left(\left(n \backslash S\right)/n\right) \; n/n \; \left(\left(n/n\right)/\left(n/n\right)\right) \left[n/n \; n\right]$$

$$n \left(\left(n \backslash S\right)/n\right) \; n/n \; \left(\left(n/n\right)/\left(n/n\right)\right) \; n \qquad (3)$$

et toute autre application des règles de réduction devient impossible. Il en résulte que la seule succession possible des règles de réduction est celle indiquée dans (2').

3.- Si comme dans (1) et (2) la réduction permet d'aboutir à une seule catégorie, appelée numérateur, on dit que la réduction est propre.

(3) n'est pas une réduction propre.

Une chaîne d'éléments de C pour laquelle existe une réduction propre est dite connexe.

- $(n/n)/(n/n)$ $\quad n/n$ est connexe (règle R_1)
- $n/n \quad (n/n)/(n/n)$ n'est pas connexe.

Grammaire catégorielle et C-grammaires

Il a été démontré (Bar-Hillel [4]) qu'une grammaire catégorielle est équivalente à une grammaire CF en ce sens qu'elles permettent de reconnaître le même langage.
Il s'en suit qu'il est impossible de construire une grammaire catégorielle qui puisse décrire toutes les sensibilités d'une langue naturelle.

Comme dans une grammaire transformationnelle, les transformations sont définies sur l'arbre syntaxique, rien n'empêche d'adjoindre à une grammaire catégorielle l'équivalent de la composante transformationnelle. C'est ce que nous faisons dans la deuxième partie du mémoire.

2.4.3. Intérêts de l'approche catégorielle

1.- Les grammaires catégorielles permettent de faire une grande économie du fait qu'elles nous dispensent de toute structure syntaxique.
Les catégories grammaticales - primaires ou secondaires - figurent dans le vocabulaire et peuvent être définies de la façon suivante [12] :

phrase :	S
nom :	n
adjectif :	n/n
verbe intransitif :	$n \backslash S$
verbe transitif :	$(n \backslash S)/n$
préposition :	$(S \backslash S)/n$
adverbe :	$(n/n)/(n/n)$
article :	n/n

Etant donné une chaîne x de mots de V, après l'affectation à chaque mot de x de la catégorie grammaticale correspondante, une simple procédure de calcul basée sur les règles R_1 et R_2, permet de construire la structure syntaxique de x (calcul syntaxique).

2.- La même procédure de calcul permet non seulement de tester la "connexite syntaxique d'une chaîne donnée d'éléments de V, mais aussi de déterminer les différents constituants de la chaîne.

Considérons par exemple les deux chaînes :

Alpha mange (6)

Le pauvre Alpha mange (7)

(6) est connexe :

$$n \quad n\backslash S \longrightarrow S$$

(7) est connexe :

$$n/n \quad n/n \quad n \quad n\backslash S \qquad (7')$$

$$\frac{\dfrac{\dfrac{n/n \quad n/n}{n} \quad n}{n} \quad n\backslash S}{S}$$

mais (7') montre que "Alpha mange" n'est pas un constituant de (7) dont les constituants sont :

$[\text{pauvre Alpha}]_n$

$[\text{Le } [\text{pauvre Alpha}]_n\ ]_n$

3.- Des programmes d'analyse syntaxique très rapides ont été construits sur la base de grammaires catégorielles (Bobrow [6]).

4.- Il faut noter toutefois quelques inconvénients

a) l'exemple (2) montre bien que d'une façon générale quelque soit le sens de lecture de la chaîne (de la gauche vers la droite ou l'inverse), il faut faire plusieurs essais avant de trouver les bonnes réductions. Ceci est dû au fait que les règles proposées ne sont pas associatives : la grammaire G_1, proposée dans la deuxième partie du mémoire évite cet inconvénient.

b) Le tableau (5) montre que la forme de catégories primaires et secondaires varie beaucoup, ce qui peut être un handicap dans la construction de la procédure de calcul.

c) du fait que les règles de réduction sont basées sur la concaténation, le système catégoriel ne permet pas de rendre compte des constituants discontinus.

3. THEORIE SEMANTIQUE DE WOODS

Nous reprenons ici les idées exprimées dans la thèse de Woods [34] et dans son article [35].

3.1. Sémantique d'un système d'interrogation

A la suite de Bobrow, Woods considère que la structure sémantique d'un système d'interrogation peut être décrite par un modèle qui représente la façon dont le locuteur appréhende le monde extérieur ("speaker's model of the world) et qui se définit de la façon suivante :

(1) un ensemble d'objets O_i représentés par des syntagmes nominaux :

exemple : un avion
AA-57

(2) un ensemble de fonctions $F_{i,n}$ qui transforment chacune un n-uple d'objets en un autre n-uple d'objets :

exemple : le père de ——;
la somme de —— et ——;

(3) un ensemble de relations $R_{i,n}$ représentées par des syntagmes verbaux suivis éventuellement de prépositions qui introduisent les compléments verbaux :

—— dort;
—— est égal à;
—— part de —— à ——;

Ces relations correspondent aux "concepts" que le système peut comprendre.

(4) un ensemble de propositions Pi qui sont des réalisations des relations sur des objets définis appelés arguments :

Exemple :

AA - 57 part de Boston à 8 h.

Les propositions correspondent aux "connaissances" du système : il sait que la relation

—— part de —— à ——;

existe entre les objets AA-57 et Boston.

(5) un ensemble de règles sémantiques qui permettent de déduire de nouvelles propositions à partir de celles qui sont déjà connues du système.

Dans tout ce qui précède les —— représentent des objets ce qui définit la nature des arguments dans ce système.

Woods a ajouté à ce système les opérateurs logiques ET, OU, ... et les quantificateurs considérés comme des prédicats dont les arguments sont des propositions : ce qui revient à élargir le système de Bobrow en disant que les arguments peuvent être aussi bien des objets que des propositions.

Exemple :

ET - un est prédicat

ET (P_1, P_2)

dont les arguments sont deux propositions P_1 et P_2.
ET (P_1, P_2) est vrai si P_1 et P_2 sont vrais.

TOUS - est un prédicat

$$p(X, P(X,...))$$

où X est un objet, P(X, ...) une proposition dans laquelle X intervient comme argument. p(X, P(X,...)) est vrai si P(X,...) est vraie pour toutes les assignations possibles de X.

Une phrase en langage naturel est alors interprétée comme un ensemble de relations qui relient certains objets. Cette interprétation est représentée explicitement comme une instruction au système.

3.2. Primitives sémantiques

On appelle primitives sémantiques du système les prédicats, fonctions et instructions que le système "comprend". On dira qu'un ensemble de primitives p_i est acceptable (adéquate) pour une base de données déterminée, si en utilisant les p_i dans des questions, il est possible de retrouver tout élément de la base de données.

Cet ensemble est relativement réduit pour un sujet déterminé : Woods construit 33 primitives sémantiques pour l'interrogation d'un horaire aérien.

On suppose que la composante de recherche contient des procédures qui définissent le sens de chaque primitive.

Exemple : la primitive CONNECT (X1, X2, X3) est un prédicat défini par une procédure appelée CONNECT avec des paramètres X1, X2, X3 qui détermine si oui ou non la proposition

CONNECT (X1, X2, X3)

est vraie lorsque X1 est un vol, X2, X3 des villes et qui

signifie que le vol X1 va de la ville X2 à la ville X3.

3.3. Langage de commande

Les primitives sémantiques étant définies, si on suppose que la composante de recherche contient les procédures qui permettent de :

- calculer la valeur de vérité de tout prédicat avec des arguments donnés
- calculer la valeur de toute fonction sur des arguments définis
- réaliser toutes les instructions primitives, le problème de l'interprétation sémantique devient indépendant de celui de la structure de la base de donnée, ou des techniques de recherches une fois qu'on a défini un langage de commande. Ce langage de commande définit la syntaxe et la sémantique des expressions que le système "comprend".

Le langage de commande contient trois constructions de base : l'instruction, la proposition et le désignatif.

1) Une expression du langage de commande est une instruction dont les arguments sont des propositions et/ou des désignatifs.

2) Il existe deux instructions de base :
TEST (P) - est l'instruction qui détermine la valeur de vérité de la proposition P
LIST (X) - est l'instruction qui imprime le nom de l'objet désigné par le désignatif X.

3) une proposition est un nom de prédicat suivi d'une liste d'arguments entre parenthèses :

DEPART (AA-57, BOSTON)

4) un désignatif est :
- soit un nom propre : AA-57, BOSTON
- soit un nom de variable : X1
- soit un nom de fonction suivi de ses arguments :
  OWNER (AA-57) - le propriétaire de AA-57

5) une variable est X suivi d'un entier : X1, X2 ...

Les propositions et les instructions peuvent être quantifiées par des quantificateurs de la forme :

(FOR QUANT X / CLASS : R(X); P(X))

où QUANT - est un "quantificateur" : every, some, the ...
X - un nom de variable
CLASS - désigne le domaine à quantifier
R(X) - éventuellement non présente, est une proposition qui indique des restrictions sur le domaine à quantifier.
P(X) - est la proposition ou l'instruction qu'on est entrain de quantifier.

Exemple

(FOR EVERY X1/ FLIGHT : DEPART (X1, BOSTON) : LIST(X1))

est une instruction quantifiée qui demande à la composante de recherche d'imprimer le nom de chaque vol qui part de BOSTON.

On suppose qu'il y a peu de domaines à quantifier et que dans chacun de ces domaines, il existe une fonction d'énumération qui permet d'énumérer tous les éléments du domaine.
Il existe trois autres types de "quantificateurs" dits numériques :

a) (FOR n MANY X / CLASS : R(X); P(X))
qui est vraie s'il existe au moins n objets X, X ∈ CLASS, et tels que R(X) et P(X) sont vraies.
Exemple
(FOR 5 MANY X1 / FLIGHT : CONNECT (X1, BOSTON, CHICAGO); LIST (X1))
correspond à la phrase :
"Donnez le nom de 5 vols qui vont de BOSTON à CHICAGO"

b) (FOR GREATER (n,M) MANY X / CLASS : R(X); P(X))
permet de spécifier une condition supplémentaire sur le nombre d'objets.
Exemple
TEST ((FOR GREATER (N,30) MANY X1/FLIGHT : JET (X1); DEPART (X1, BOSTON)))
correspond à la phrase :
"Est-ce que plus de 30 jets quittent BOSTON ?"

c) NUMBER (X / CLASS : R(X))
est une fonction qui retourne le nombre d'objets X, X ∈ CLASS, tels que R(X) est vraie.
Exemple
LIST (NUMBER (X1 / FLIGHT : CONNECT (X1, BOSTON, CHICAGO)))
traduit la phrase :
"Combien de vols vont de Boston à Chicago ?"

On notera que ce langage est en gros équivalent à un calcul de prédicats du premier ordre avec des quantificateurs plus élaborés et l'addition d'instructions que l'on ne peut exprimer en logique du premier ordre.

3.4. Interprétation sémantique

Le but de l'interpréteur sémantique est de traduire l'arbre syntaxique de la phrase en une représentation formelle de son sens en fonction des primitives sémantiques (expression du langage de recherche). Ce processus doit être fini pour que la phrase ait un sens. Ainsi, l'interpréteur doit décomposer l'arbre syntaxique en sous-arbres dont il connaît l'interprétation. L'interprétation de la phrase sera alors une composition des interprétations des sous-arbres.

Le fonctionnement de l'interpréteur est défini par un ensemble fini de règles sémantiques :

figure $\Longrightarrow$ action
(pattern $\Longrightarrow$ action)

où 'figure' est la description d'une sous-structure et 'action' l'interprétation de cette sous-structure.

Pour la détermination de cette sous-structure, on distingue fondamentalement deux types de noeuds dans l'arbre syntaxique

- le noeud S - qui correspond à l'interprétation de la phrase
- les noeuds NP - qui correspondent aux syntagmes nominaux

L'interprétation sémantique de S est soit une proposition soit une instruction tandis que l'interprétation de NP

est un désignatif. Ainsi l'interpréteur est divisé en deux interpréteurs correspondant à ces deux types de noeuds :

- le S - processeur qui interprète les noeuds S
- le NP - processeur qui interprète les noeuds SNP et peut produire des quantificateurs qui influencent toute l'interprétation de la phrase.

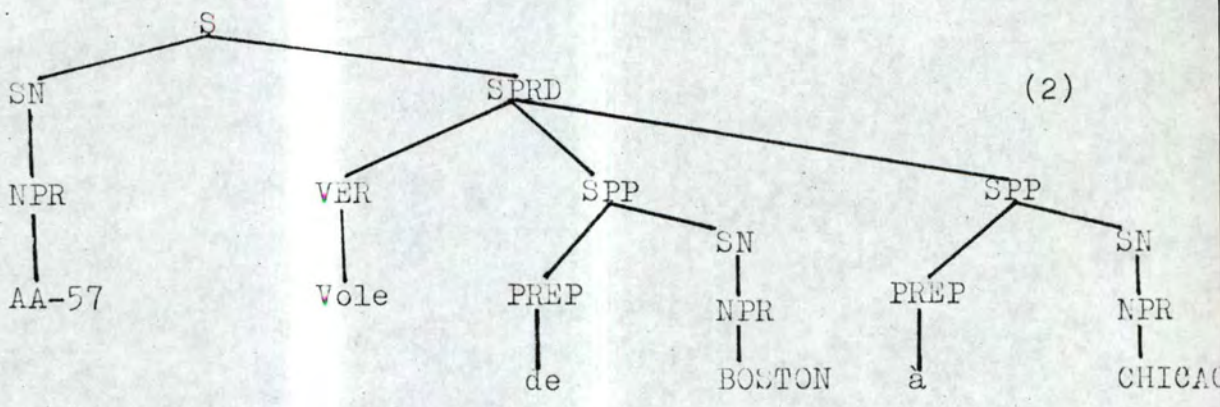
3.5. Le S-processeur

En supposant que les noms propres sont directement interprétables, on peut illustrer le fonctionnement du S-processeur indépendamment du NP-processeur en considérant des phrases dont les syntagmes nominaux se réduisent à des noms propres.

Soit par exemple la phrase :

AA-57 vole de BOSTON à CHICAGO (1)

dont l'arbre syntaxique est le suivant :



Comme en général, les verbes correspondent en gros aux prédicats, le verbe sera le premier élément de l'arbre syntaxique qui intervient dans la détermination de la primitive (ou composition de primitives). Ainsi dans (2) la primitive sera le prédicat CONNECT qui correspond au verbe `voler'. Ensuite, le S-processeur doit vérifier que tous les arguments du prédicat CONNECT sont présents dans l'arbre syntaxique et qu'ils sont valides. Dans ce cas-ci, il est nécessaire que le sujet soit un vol, les deux compléments verbaux des noms de ville introduits respectivement par les prépositions `de' et `à'. Ces tests permettent d'écarter des phrases sémantiquement mauvaises comme :

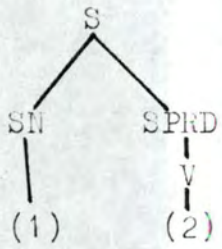
BOSTON vole de AA-57 à CHICAGO.

Le S-processeur doit donc être à même de poser certaines questions sur les constituants de la phrase : catégorie grammaticale, fonction grammaticale, domaine, ...

Les relations grammaticales entre les éléments de l'arbre syntaxique sont définies par des sous-arbres.

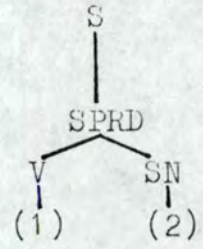
Exemple

G1 :

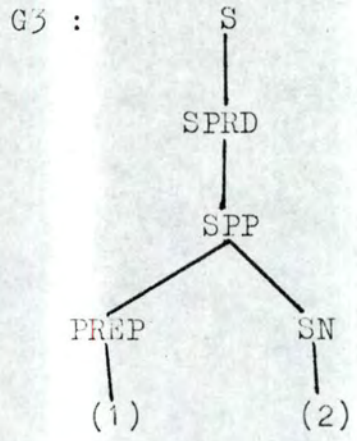


sujet-verbe

G2 :



verbe-complément d'objet direct



syntagme prépositionnel
complément verbal.

3.6. Structure des règles sémantiques

La figure ou partie gauche d'une règle sémantique est définie en termes de schéma (template). Le schéma est un sous-arbre (comme G1, G2, G3) muni des conditions sur les noeuds terminaux.

Les conditions sont exprimées par des primitives sémantiques et la relation 'égale':

$$(n) = \text{`}\ W\ \text{'}$$

qui signifie que le noeud terminal n est identique à la chaîne de caractères W.

A l'intérieur d'une figure, chaque schéma est identifié en plus par un entier, utilisé dans la partie droite - action, pour référencer les noeuds numérotés :

le noeud n-m indique le mè noeud du nè schéma.

Exemple

1- (G1 : VOL ((1)) et (2) = vole) et
2- (G3 : (1) = de et LIEU ((2)) et
3- (G3 : (1) = à et LIEU ((2))

CONNECT (1-1, 2-2, 3-2)

qui s'interprète de la façon suivante :

1) Si le noeud en considération A est la racine d'un sous-arbre de la forme G1 et dont le noeud correspondant à (1) indique un vol, le noeud correspondant à (2) domine directement le mot `voler´.

2) Si A est la racine d'un sous-arbre de la forme G3 avec le noeud correspondant à (1) dominant directement le mot de et le noeud (2) désignant un lieu

3) Si A est la racine d'un autre sous-arbre de la forme de G3 avec le noeud (1) dominant directement le mot `à´ et le noeud (2) désignant un lieu alors l'interprétation de A est

CONNECT (1-1, 2-2, 3-2)

Ceci appliqué à l'arbre (2) donne

CONNECT (AA-57, BOSTON,CHICAGO)

Les règles sémantiques du S-processeur sont dites S-règles.
Dans le dictionnaire, chaque verbe figure avec l'ensemble des S-règles qu'on peut lui appliquer.

Les règles sémantiques contiennent des conditions sémantiques de la forme :

VOL ((1)), LIEU ((2))

Cela signifie que le dictionnaire doit indiquer la classe sémantique des syntagmes nominaux. Pour un nom propre ce sont les ensembles auxquels appartient l'objet désigné :

AA-57 / VOL
BOSTON / VILLE, LIEU

Pour un nom commun, on peut indiquer les ensembles qui contiennent la classe des objets désignés :

vol / VOL
ville / VILLE, LIEU

On peut aussi définir des conditions sémantiques plus complexes par des routines que l'interpréteur peut appeler.

3.7. Le NP-processeur

Le fonctionnement du NP-processeur est analogue à celui du S-processeur ; il est défini par des règles sémantiques de même forme: ici dans les schémas, les sous-arbres ont pour racine un SN (au lieu de S dans les S-règles)

Le processus d'interprétation d'un syntagme nominal se fait en trois étapes :

- détermination du quantificateur utilisé
- détermination du domaine à quantifier
- détermination des restrictions de quantification

Toutes ces déterminations se font au moyen de règles sémantiques qui ont les mêmes structures générales que les S-règles.

figure ⟹ action

Seule la forme de la partie action peut être considérée comme caractéristique.

1) Le quantificateur est déterminé au moyen de D-règles dont l'action est de la forme :

$$\Longrightarrow (\text{FOR QUANT X / } \nabla \text{ ; } \Delta) \qquad (5)$$

Les symboles Δ et ∇ sont des chaînes de travail qui indiquent les positions où seront insérés respectivement :

- l'interprétation sémantique de la phrase
- le domaine à quantifier et ses restrictions

2) le domaine lui est déterminé par le nom qui figure dans le syntagme nominal au moyen des N-règles dont la partie droite est de forme générale

$$\text{CLASS : } \nabla \qquad (6)$$

On suppose qu'à chaque domaine est associée une fonction d'énumération. Le symbole ∇ indique la position où seront insérées les restrictions possibles.

3) Les restrictions peuvent être définies par une phrase relative modifiant un syntagme nominal. Dans ce cas, le NP-processeur appelle le S-processeur qui interprète la phrase relative et cette interprétation est la restriction cherchée.

Mais l'interprétation des adjectifs et compléments nominaux se fait au moyen de R-règles. Les parties droites de ces règles sont des propositions qui doivent être ajoutées comme contraintes au domaine de quantification.

A l'issue de cette troisième étape, le NP-processeur a déterminé toutes les restrictions indiquées par les R-règles et les phrases relatives. Il les compose en une proposition qui remplace le symbole ∇ . S'il n'y a pas de restriction, ∇ est remplacé par un — (restriction vide).

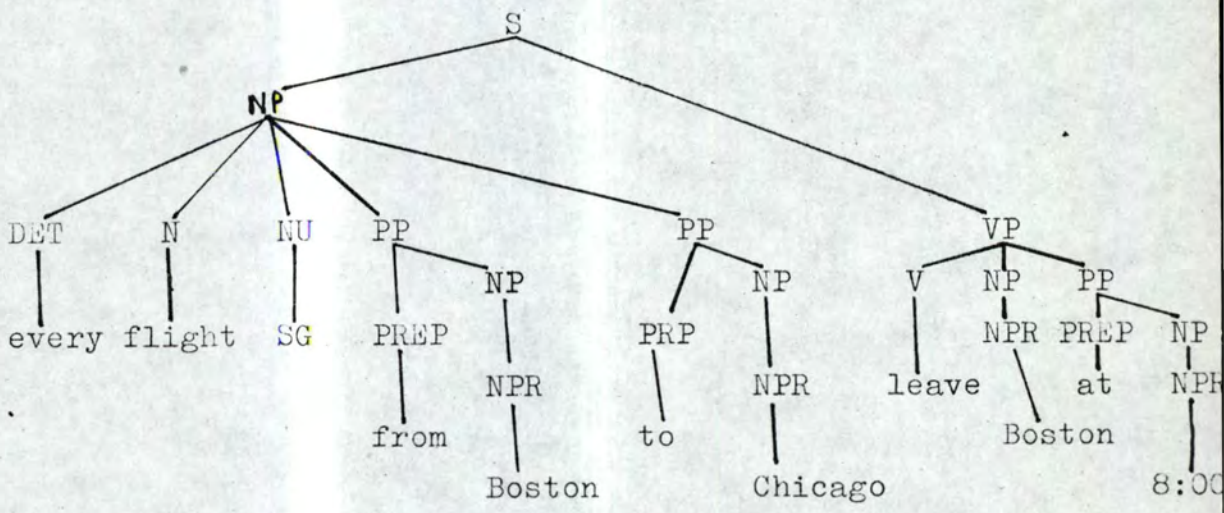
3.8. L'interpréteur de Woods

Nous allons examiner sur un exemple pratique le fonctionnement logique de l'interpréteur sur la base de ce qui a été dit plus haut.

Soit la phrase :

"every flight from Boston to Chicago leaves Boston at 8:am"

dont l'arbre syntaxique est le suivant :



1) En analysant S, le S-processeur commence par créer une zone de travail W et l'ini tialise avec

$$W := \Delta$$

Ensuite, le S-processeur appelle le NP-processeur pour que celui-ci interprète le syntagme nominal

every flight from Boston to Chicago

2) Le NP-processeur crée une nouvelle variable qui n'a pas encore été utilisé - en concaténant X et un entier ici 1 et associe cette variable X1 au syntagme nominal.

3) Détermination du quantificateur. Le NP-processeur trouve le déterminant du syntagme nominal (ici every) et cherche la D-règle correspondante, D2. Il met dans une zone de travail z la partie droite de cette règle :

$$Z := (\text{FOR EVERY X1} / \nabla ; \Delta)$$

4) Détermination du domaine à quantifier. Elle se fait par le nom `flight' dont le dictionnaire fournit les N-règles. Le NP-processeur trouve la N-règle qui s'applique - ici N1 dont la partie droite :

$$\text{FLIGHT} : \nabla$$

va remplacer dans Z le symbole ∇ :

$$Z := (\text{FOR EVERY X1} / \text{FLIGHT} : \nabla ; \Delta)$$

5) Détermination des restrictions. Ces restrictions sont données par les deux compléments nominaux :

from Boston
to Chicago

Le NP-processeur essaie les différentes R-règles de `flight' qui sont données dans le dictionnaire pour trouver la bonne R5 dont la partie droite

$$\text{CONNECT (X1, BOSTON, CHICAGO)}$$

est mis à la place de ∇ dans Z pour donner :

Z : = (FOR EVERY X1 / FLIGHT : CONNECT (X1, BOSTON, CHICAGO); Δ)

6) Le quantificateur est ainsi généré et le NP-processeur retourne la valeur z au S-processeur qui remplace dans sa zone de travail W, Δ par Z :

W : = (FOR EVERY X1 / FLIGHT : CONNECT (X1, BOSTON, CHICAGO); Δ)

7) Le S-processeur interprète ensuite le syntagme prédicatif :

leave Boston at 8:00 a.m.

L'entrée de `leave' dans le dictionnaire fournit les différentes S-règles qui peuvent s'appliquer à `leave'. Le S-processeur trouve la règle qui convient S9 dont la partie droite fournit l'interprétation :

EQUAL (DTIME (X1, BOSTON), 8:00 a.m.)

qui est finalement substitué à Δ dans W :

W : (FOR EVERY X1 / FLIGHT : CONNECT (X1, BOSTON, CHICAGO);
EQUAL (DTIME (X1, BOSTON), 8:00 a.m.))

qui est l'interprétation sémantique de la phrase.

En pratique l'interpréteur fonctionne sur la base de 36 S-règles, 16 D-règles, 17 N-règles et 11 R-règles.

II. LES BASES D'UN SYSTEME D'INTERROGATION - NATLNG

Nous proposons dans cette partie un langage naturel d'interrogation et les bases d'un système d'interrogation qui s'appuie sur ce langage.

Pour des raisons pratiques d'exposé, la grammaire catégorielle de reconnaissance est présentée telle qu'elle a été implémentée : les catégories grammaticales sont représentées comme des vecteurs d'entiers et le calcul syntaxique qui résulte de la règle de réduction est une simple arithmétique entière. Mais la règle de réduction pourrait être traduite autrement (par exemple manipulations de pointeurs, de listes, ...)

4 . LE LANGAGE NATUREL

4.1. Domaine du langage

Pour illustrer le fonctionnement du système d'interrogation qui a été défini, nous allons supposer que le langage d'interrogation porté sur un catalogue de bibliothèque qui sera considéré comme la base de donnée B qu'utilise le système.

Nous ne poserons pas ici le problème de la définition exacte du contenu et de l'organisation de B, le but cherché étant justement de définir un système qui en soit indépendant.

Le catalogue est donné sous la forme d'un tableau qui contient les objets suivants : AUTEUR, TITRE, COLLECTION, EDITEUR, ANNEE, COTE, FORMAT.

Exemple

AUTEUR	:	MARCUS, S.
TITRE	:	Algebraic Linguistics; Analytical Models
COLLECTION	:	Mathematics in Science and Engineering, V29
EDITEUR	:	Academic-Press
ANNEE	:	1967
COTE	:	960/675
FORMAT	:	XIV - 258 p., 8°

Le langage naturel, LN, est l'ensemble des phrases admises qui permettent de nommer ces différents objets et de poser des questions sur leurs relations (que nous supposons traduites dans B).

4.2. Types de phrases

Toutes les phrases de LN auront la même forme que celle qu'elles ont habituellement dans la langue ordinaire.

Nous caractériserons le type de phrases par deux notions :

- la forme : déclarative ou interrogative
- le signe : positif ou négatif

La structure des phrases positives est la suivante :

Phrases déclaratives

Le type le plus général d'une phrase déclarative est :

SUJ PRED COD CPV CPX .

1) le sujet - SUJ : il comprend obligatoirement un syntagme nominal suivi facultativement soit :

- d'un syntagme prépositionnel dit complément nominal (CPN) :

la thèse (de "Woods")$_{CPN}$ ——

- d'une apposition qui est nécessairement un nom d'un objet de B :

le livre "Automated Language Processing" ——

2) le prédicat - PRED : il comprend un verbe conjugué au présent suivi éventuellement d'un adjectif. La distinction n'est pas faite entre verbes et copules

—— (est disponible)$_{PRED}$ ——

3) le complément d'objet direct = COD : c'est un syntagme nominal suivi éventuellement d'une phrase relative introduite par que ou qui :

Vous avez la thèse de "Woods".

Donnez la liste des livres de "Chomsky" que vous avez .

4) les compléments verbaux = CPV : ce sont des syntagmes prépositionnels qui suivent un prédicat qui n'a pas de compléments d'objet direct.

Lorsque le prédicat a déjà un complément d'objet direct, les syntagmes prépositionnels sont considérés comme des compléments indéfinis (CPX) et c'est l'interpréteur qui permettra de dire si :

- ce sont des compléments nominaux qui se rattachent au complément d'objet direct :

vous avez le livre (de "Marcus")$_{CPX}$

-ou bien si ce sont des compléments verbaux qui se rattachent au prédicat :

vous avez le livre de "Marcus" (dans la salle de lecture)$_{CPX}$

Une phrase contient au plus deux propositions et dans chaque proposition il y a au plus deux compléments nominaux, deux compléments verbaux et deux compléments indéfinis.

Phrases interrogatives

Les types de phrases interrogatives du langage sont les suivants :

1) les phrases introduites par le pronom interrogatif 'qui' suivi immédiatement du prédicat :

Qui est l'auteur de —— ?

2) Les phrases introduites par les différentes formes de l'adjectif interrogatif 'quel' :

Quels sont vos ouvrages relatifs à —— ?
Quel est le titre du dernier livre de —— ?

3) Les phrases introduites par la formule 'est-ce que' suivie du sujet de la phrase :

est-ce que vous avez —— ?

4) les phrases interrogatives construites directement à partir de phrases affirmatives qui contiennent un complément d'objet direct :

avez-vous le livre —— ?

Par contre la phrase :

le livre —— est-il disponible ?

n'est pas permise.

Phrases négatives

Une seule négation est permise par phrase et elle est du type :

—— ne —— pas ——

Remarques :

1) les interrogations introduites par quand, où, pourquoi, comment, combien peuvent toujours se ramener à des interrogations introduites par quel :

quand - à quel moment
où - à quelle place, dans quelle direction
pourquoi - pour quelle raison
comment - de quelle manière
combien - en quelle quantité, ...

2) les seules phrases impératives admises sont introduites par écrivez et donnez

3) les connecteurs (`,´ , `;´ , `et´, `ou´) ne sont pas permis.

5 . GRAMMAIRE DE RECONNAISSANCE

L'analyse syntaxique est basée sur une grammaire catégorielle :

$$G_1 = (V, C, S, R, f) \qquad (1)$$

dont nous allons définir les différents éléments.

5.1. Le vocabulaire

Le vocabulaire V comprend deux parties :

1) un vocabulaire grammatical V_G qui est l'ensemble des éléments qui interviennent dans la formulation des commandes et qui ne désignent pas des objets de la base de donnée.
Ce sont :

- les articles : le, la, les, l',...
- les prépositions : de, à, par, sur, ...
- les adverbes : très, peu, non, ne, ...
- les pronoms personnels : je, il, elle, nous, ...
- les adjectifs : premier, libre, ...
- les verbes : emprunter, écrire, publier, ...
- les copules : avoir, être
- les noms communs : nom, livre, revue, ouvrage, auteur, éditeur, ...
- les signes : trait d'union, virgule, point, ...

2) un vocabulaire V_{BD} qui est l'ensemble des éléments qui désignent les objets de la base de donnée :

"KNUTH","DIEUDONNE", ..., —— noms d'auteurs
"C.A.C.M", "J.A.C.M.",... —— noms de revues

Nous supposerons que V_{BD} et V_G forment une partition de V :

$$V_G \cap V_{BD} = \emptyset$$
$$V = V_G \cup V_{BD}$$

En pratique V est choisie de façon à pouvoir d'une part désigner tous les objets de B, d'autre part exprimer l'ensemble des relations de B. Un exemple est donné en annexe (A.1.)

5.2. Les catégories grammaticales

C'est l'ensemble des catégories Ci :

$$C_i = CAT_i\ (VE_i, VS_i, IND_i)\ i = 1, 2, ..., m$$

définis comme suit :

1) CAT_i est le nom de la catégorie C_i. Nous avons repris les mêmes noms que ceux utilisés dans les grammaires CF.

ART - article
PREP - préposition
SN - syntagme nominal
SVER - syntagme verbal
SPP - syntagme prépositionnel

2) VE_i - la première composante est dite valeur d'entrée de C_i. C'est le numéro d'ordre i de C_i dans c.

3) VS_i - est un entier positif ou nul appelé valeur de sortie de C_i - S_i VS_i est positif, C_i est dite positive sinon C_i est nulle.

4) la troisième composante IND_i est l'indicateur de C_i et peut être l'un des signes suivants : /, |, \.

C_i est dite respectivement centrale, gauche, droite selon que IND_i est égal à |, /, \.

Exemples :

C_1 = PREP (1, 1, /) préposition - catégorie positive gauche
C_4 = SN (4,4,|) syntagme nominal - cat. positive centrale
C_5 = SPP (5, 0, \) syntagme prépositionnel - cat. nulle droi
C_7 = ART (7, 0, /) article - catégorie nulle gauche
C_{10} = VER (10, 10, |) verbe -catégorie positive centrale.

La catégorie distinguée est la catégorie S.

$$S_{15} = S(15, 0, \backslash)$$

C est donnée en annexe (A, 2)

5.3. Réductions

C_i et C_j étant deux éléments de C :

$$C_i = CAT_i\ (VE_i, VS_i, IND_i)$$
$$C_j = CAT_j\ (VE_j, VS_j, IND_j)$$

Nous dirons que C_i réduit C_j dans une chaîne donnée d'éléments de C où C_i et C_j sont consécutifs si :

a) IND_i = / et IND_j = / ou bien
IND_j = | ou bien
IND_j = \

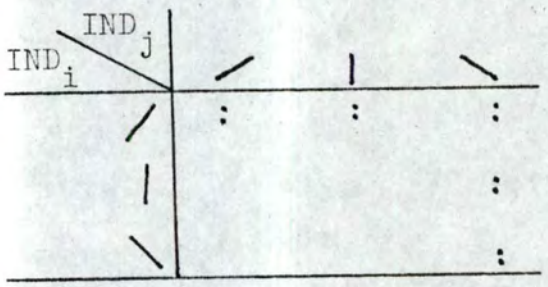
b) IND_i = | et IND_j = \

c) IND_i = \ et IND_j = \

et nous noterons ce fait par deux points entre C_i et C_j :

$$C_i : C_j$$

On peut résumer a, b, c dans le tableau suivant :



(2)

Nous dirons par la suite ' points de réduction (') et tableau de réduction (2)

Exemple : soit la chaîne

$$C_1\ C_7\ C_4\ C_{10}\ C_5\ C_1\ C_7 \qquad (3)$$

C_1 = prep (1,1, /) réduit C_7 = ART (7,0,/)
C_7 = ART (7, 0, /) réduit C_4 = SN (4, 4, |)
C_{10} = VER (10, 10, |) réduit C_5 = SPP (5, 0, \)

ce que nous pouvons noter :

$$C_1 : C_7 : C_4 \quad C_{10} : C_5 \quad C_1 : C_7$$

5.3.1. Remarques

Les deux lectures possibles de la table (2) :

ligne-colonne ou colonne-ligne

correspondent respectivement aux deux sens de lecture possible d'une chaîne :

Lecture de la gauche vers la droite ou l'inverse.

Par ailleurs nous noterons que :

1) une catégorie droite ($IND_i = \backslash$) ne peut réduire qu'une catégorie droite ($IND_j = \backslash$)

2) une catégorie centrale ($IND_i = |$) ne peut réduire qu'une catégorie droite ($IND_j = \backslash$)

3) une catégorie gauche peut réduire toute autre catégorie.

5.3.2. Règle de réduction R

Pour toute suite, C_1, C_2 , ... , C_i telle que :

$$C_1 : C_2 : \ldots : C_i$$

nous définissons une règle de réduction R :

$$R\ (C_1, C_2, \ldots, C_i) = C_{VS_1 + VS_2 + \ldots + VS_i} = C_K \qquad (4)$$

C_K est le numérateur de C_1, C_2, ..., C_i que nous appellerons intervalle.

En d'autres termes, un intervalle est l'ensemble des catégories consécutives qui se réduisent à une même catégorie dite numérateur. Et R signifie que la valeur d'entrée de C_K (son indice K), est la somme des valeurs de sorties des catégories C_1, ..., C_i.

5.3.3. Propriétés de R

1) R n'est pas commutative.

Considérons par exemple le couple :

$$C_7 = \text{ART}\ (7, 0, /) \quad \text{et} \quad C_4 = \text{SN}\ (4, 4, |)$$

D'après le tableau (2), $C_7 : C_4$ alors que l'inverse n'est pas vrai. Ce qui traduit le fait que 'le livre' est un syntagme nominal alors que 'livre le' n'en est pas un.

Donc R (C_7, C_4) est définie mais R (C_4, C_7) ne l'est pas et R n'est pas commutative.

2) R est associative.

Soit C_1, C_2, C_3 telles que $C_1 : C_2 : C_3$.

Alors du fait que :

$$VS_1 + VS_2 + VS_3 = (VS_1 + VS_2) + VS_3 = VS_1 + (VS_2 + VS_3)$$

$$R\left[R(C_1, C_2), C_3\right] = R\left[C_1, R(C_2, C_3)\right] = R(C_1, C_2, C_3).$$

Nous notons à ce niveau :

$$R(C_1, C_2) : C_3$$

vient du fait que $C_2 : C_3$ et non de la catégorie numérateur de C_1 et C_2.

Cette associativité signifie qu'en pratique les réductions R peuvent se faire dans n'importe quel ordre.

3) il résulte du tableau (2) et des remarques (5.3.1.) que la forme la plus générale d'un intervalle I est une suite :

$$x_1, x_2, \ldots, x_p \quad b \; y_1, y_2, \ldots y_q$$

où les n_i sont des catégories gauches, b- une catégorie centrale, y_j - des catégories droites.

Un tel intervalle qui contient une catégorie centrale est dit <u>significatif</u>.

Si I est un intervalle contenant une catégorie C_i, le complémentaire de C_i par rapport à I est le <u>voisinage</u>. $v(C_i)$ de C_i.

<u>Exemple</u> : dans (3)

$C_1 \; C_7 \; C_4$ est un intervalle significatif

$$v(C_1) = \{C_7, C_4\}$$
$$v(C_7) = \{C_1, C_4\}$$
$$v(C_4) = \{C_1, C_7\}$$

C_1C_7 est un intervalle non significatif. Les intervalles significatifs seront en fait les arguments du réseau grammatical élargi qui sera introduit en (6.3.).

5.4. <u>Fonction d'assignation</u>

La fonction d'assignation f est une fonction qui permet de faire correspondre à chaque élément x_i d'une suite

$$x_1, x_2, \ldots x_n$$

d'éléments de V, une catégorie grammaticale :

$$f : x_i \longrightarrow C_K \in C$$

f est définie de la façon suivante :

1) les éléments de V_G

a. les catégories grammaticales ordinaires correspondent de la façon suivante aux catégories C_i :

préposition	- PREP (1, 1, /)	nom propre	- NPR (20, 4, |)
adverbe	- ADV (2, 0, /)	nom commun	- N (21, 4, |)
article	- ART (7, 0, /)	pronom	- PRO (22, 4, |)
verbe	- VER (10, 10, |)		

Les adjectifs sont répartis en deux catégories .

ADJ1 (8, 0, /)

ADJ2 (9, 0, \)

selon qu'ils se placent respectivement à gauche ou à droite du nom qu'ils qualifient.

Les catégories suivantes :

auxiliaire - AUX (3, 1, /)

copule - COP (12, 10, /)

modalité - MOD (13, 0, /)

déterminant - DET (16, 0, /)

ont été retenues en vue d'un développement ultérieur du système.

b. catégories particulières :

- adverbe de négation

f(`ne´) = NEG (18, 18, |)

f(`pas´) = NEG2 (19, 0, \)

- catégorie "connecteur"

f (`et') = f (`ou') = f (`,') = CONN (24, 0, 1)

- catégorie "entier"

nombre entier - INT (23, 4, 1)

- adjectifs, adverbes interrogatifs et pronoms interrogatifs :

f (`quel') = f (`quelle') = f (`quels') = f (`quelles')
= f (`qui') = Q1 (25, 0, 1)
f (`combien') = Q2 (26, 0, 1)

- trait d'union

f (`-') = INV (32, 0, 1)

- pronoms relatifs

f (`que') = RQUE (27, 0, 1)
f (`qui') = RQUI (28, 0, 1)

- fin de phrase

f (`.') = f (`?') = FIN (32, 0, 1)

2) tous les éléments de V_{BD} sont classés dans la catégorie nom propre.

$x \in V_{BD}$ f (x) = NPR (20, 4, 1)

3) les catégories qui peuvent résulter des réductions et peuvent être des numérateurs :

a. le syntagme nominal : SN (4, 4, 1). C'est le numérateur d'un intervalle significatif dont la catégorie positive est une des catégories :

nom propre NPR (20, 4, 1)
nom commun N (21, 4, 1)
pronom personnel PRO (22, 4, 1)
nombre entier INT (23, 4, 1)

b. le syntagme prépositionnel SPP (5, 0, \). C'est le numérateur d'un intervalle significatif dont deux catégories sont positives : une préposition PREP (1, 1, /) et son voisinage qui est un syntagme nominal SN (4, 4, 1).
SPP est une catégorie nulle droite.

c. SVER - syntagme verbal. C'est le numérateur d'un intervalle significatif dont la seule catégorie positive est un verbe

SVER (11, 10, /)

est une catégorie positive gauche.

d. le syntagme prédicatif - SPRD (14, 11, \) est une catégorie positive droite. C'est le numérateur d'un intervalle significatif qui contient un syntagme verbal

e. la catégorie distinguée S déjà définie.

5.5. Stratégie de construction de C

L'idée de base est de considérer qu'une catégorie C_1 qui réduit une catégorie C_2 est un opérateur qui transforme C_2 en une autre catégorie C_3.

$$C_3 = R\ (C_1,\ C_2)$$

Nous allons examiner comment les différentes composantes des C_i sont choisies pour traduire et expliquer cette transformation.

1) nous avons repris les noms habituellement donnés aux catégories grammaticales dans les grammaires CF.

2) détermination des indicateurs. Nous partons de deux catégories fondamentales le verbe VER et le nom NCO qui sont considérées comme centrales (IND = |)

a) Le nom propre et le pronom personnel sont assimilés à la même catégorie et ont les mêmes caractéristiques que len
Le copule est assimilé à la catégorie verbe.

b) Nous rangeons ensuite les différents déterminants en deux classes : les déterminants pré posés et les déterminants post posés.

Les déterminants pré posés qui se placent à gauche de la catégorie qu'ils déterminent sont définis comme catégories gauches. Ce sont par exemple :

- les prépositions : à, de, sur, ...
- les articles : le, la, les, ...
- certains adverbes employés devant des adjectifs : très, peu, ...
- différentes catégories d'adjectifs qualificatifs ou déterminatifs : nouveau, petit, premier, vos, ...

Les déterminants post posés sont des adjectifs comme gauche, droite, rouge, ... et constituent des catégories droites (IND = /)

Les modalités (pouvoir, vouloir) sont considérées comme des déterminants pré posés au verbe et les participes passés des déterminants post posés au verbe.

3) La valeur d'entrée est un pointeur qui sert à référencer chaque catégorie dans C.

4) Détermination des valeurs de sortie. La valeur de sortie est en fait un poids qui permet de mesurer la façon dont une catégorie transforme les catégories de son voisinage.

Si C_1 réduit C_2 en une catégorie C_3 de même type que C_2, nous définissons C_1, comme nulle : l'article réduit un syntagme nominal en syntagme nominal; l'adverbe réduit un adjectif en adjectif.

Mais par contre, une préposition réduit un syntagme nominal en syntagme prépositionnel; un nom réduit un adjectif en un nom; un verbe réduit un adjectif en verbe. Ainsi, la préposition, le nom, le verbe ont des valeurs de sortie positives.

Ensuite pour que la définition de la règle de réduction soit correcte, les valeurs d'entrées des catégories qui peuvent être des numérateurs (SN, SPP, SVER, SPRD, S) sont prises égales à la somme des valeurs de sorties de leurs constituants respectifs.

Par exemple, la valeur d'entrée de SPP (5), est la somme de la valeur de sortie de PREP (1) et de SN (4).

5) les catégories spéciales NEG, Q1, Q2, Q3, INV, RQUE, RQUI sont des catégories centrales introduites pour la composante élargie et serviront à indiquer entre autres :

- le type de la phrase : interrogative, affirmative
- le signe de la phrase : positive, négative, ...

5.6. Ambiguïtés grammaticales

Il peut arriver que des mots aient des fonctions syntaxiques différentes selon le contexte dans lequel ils se trouvent. Cela revient à dire qu'il existe des mots qui ont plusieurs catégories grammaticales à la fois - une d'entre elles étant à choisir en fonction du contexte dans lequel le mot apparaît.

Nous traitons deux cas d'ambiguïtés qui ont une importance dans le langage défini :

1) "qui" - peut être pronom relatif ou interrogatif. S'il apparaît à la suite d'un syntagme nominal, il est considéré comme pronom relatif. Sinon, il doit apparaître en début de phrase et est considéré comme pronom interrogatif.

2) Certaines prépositions comme "de", "sur", peuvent être préposition nominale (introduisant un complément nominal) ou préposition verbale (introduisant un complément verbal). Le traitement est fait de la façon suivante :

- une préposition qui apparaît à la suite d'un complément d'objet direct est considérée comme indéfinie, c'est-à-dire introduisant un complément dit indéfini (CPX) et l'interpréteur sémantique permettra de dire s'il s'agit d'un complément nominal (au complément d'objet direct) ou d'un complément verbal (au prédicat).

Par exemple dans les phrases :
Vous avez le livre de Chomsky sur la table,
Vous avez le livre de Chomsky sur "les grammaires formelles".
"de" et "sur" sont considérés comme prépositions indéfinies. Alors que "de" dans ce livre traite de "programmation linéaire" est tout de suite considéré comme préposition verbale.

Les autres cas d'ambiguïté sont indiqués dans le dictionnaire comme trait distinctif des mots ambigüs pour un traitement ultérieur :

1. verbe - substantif : "programme", "analyse"
2. adjectif - substantif : "disponible","relative"
3. article - pronom : "le","la", "les", ...

6 . ANALYSE SYNTAXIQUE

6.1. Considérations générales

Dans la majorité des systèmes passés en revue par Simmons [28], on utilise des méthodes "top-down" dans l'analyse syntaxique. Cependant, comme Simmons le fait remarquer, une méthode "bottom-up" est nécessairement plus économique surtout si les constructions permises sont variées. Ce qui est le cas des langues naturelles.

Considérons par exemple l'analyse d'un syntagme nominal. Il y a au moins trois constructions permises :

NPR

PRO

ART ADJ1 N ADJ2

Dans une méthode "bottom-up" l'utilisation de la chaîne d'entrée élimine directement deux des trois possibilités alors que dans une méthode "top-down" pure toutes les trois possibilités peuvent être examinées avant de trouver la construction utilisée.

Nous utilisons dans ce qui suit une approche "bottom-up".

6.2. Structure partielle

La grammaire G_1 permet une reconnaissance très simple des phrases affirmatives. Considérons l'exemple suivant :

vous avez le dernier numéro de "C.A.C.M." (1)

Pour l'instant, nous ignorons le point qui termine la phrase.

1) La fonction d'assignation f permet de sélectionner les catégories grammaticales (C_i) correspondant aux différents mots de (1) :

PRO VER ART ADJ1 N PRP NPR (2)

dont nous rappelons la définition :

PRO (22, 4,), VER (10, 10,), ADJ1 (8, 0, /)
ART (7, 0, /), PREP (1, 1, /), NPR (20, 4,)

L'examen de la table de réduction (5.3.) permet d'écrire dans (2) les points de réduction :

PRO VER ART : ADJ1 : N PREP : NPR (3)

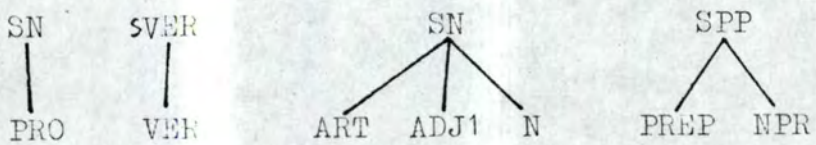
Il se dégage donc 4 intervalles significatifs :

PRO
VER
ART ADJ1 N
PREP NPR

La règle de réduction R permet de déterminer les numérateurs de ces intervalles qui sont respectivement :

SN, SVER, SN, SPP

A ce niveau, nous avons la structure suivante :



(4)

2) On peut répéter ce processus à partir de la nouvelle suite de catégories obtenues : SN, SVER, SN, SPP.

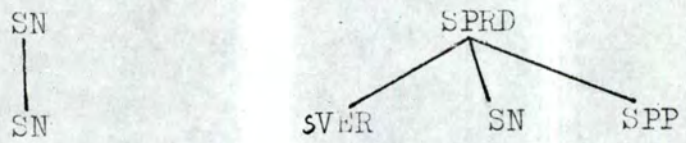
Et puisque :

SN (4, 4,), SVER (10, 10,), SPP (5, 0,)

les points de réduction s'écrivent :

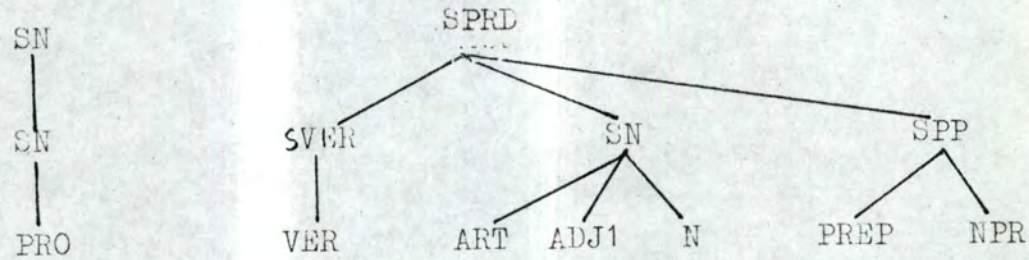
SN SVER : SN : SPP

et R permet de calculer les nouveaux numérateurs :



(5)

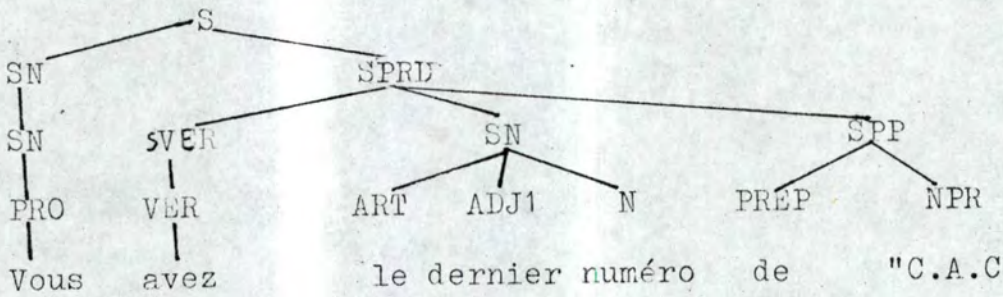
A ce niveau, nous avons la structure suivante :



3) une dernière répétition du même processus permet de faire la dernière réduction nécessaire. En effet, puisque SPRD (14, 11,\) :

SN : SVER

et S(15, 0,\) est le numérateur de cet intervalle.
D'où la structure complète de (1) :



(6)

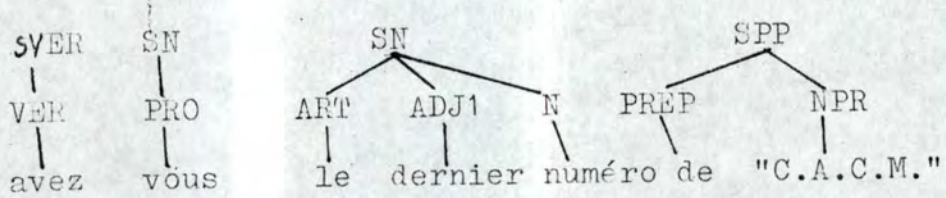
Ainsi en utilisant G_1, on peut construire (6) par un algorithme d'analyse de "bas-en-haut", et une lecture de gauche à droite par exemple.

Cet algorithme construit l'arbre (6) par balayage.

Considérons maintenant la forme interrogative de (1)

avez-vous le dernier numéro de "C.A.C.M." ? (7)

Du fait de l'inversion de "vous", la procédure ci-dessus n'est plus applicable. Après la première réduction :



(7)

il est nécessaire de signaler que le SN -"vous" ne fait pas partie du syntagme prédicatif construit avec "avez", mais doit être considéré comme le sujet de "avez".

Si nous appellons structure partielle les structures de la forme (4) et (7) obtenues après la première réduction, il nous faut donc définir une grammaire élargie comme elle a été définie dans (2.2.) qui s'applique sur cette structure partielle.

Nous donnons cette grammaire sous la forme d'un réseau grammatical élargi.

6.3. Réseau grammatical élargi

Nous utilisons le formalisme habituel (Backus-Nauer) pour définir le réseau grammatical élargi. Le trait / sépare les alternatives possibles d'une construction et l'opérateur (*) est l'étoile de Kleene qui indique un constituant répété de façon indéfinie. Le point "." indique une concatécration.

< réseau élargi >:: = (<ensemble d'arcs > <ensemble d'arcs >*)
< ensemble d'arcs > :: = (< nom d'état > <arc >*)
<arc > :: = (CAT <catégorie grammaticale > <condition >*
<action >* <sortie >)
< condition > :: = (TEST1 < catégorie grammaticale > < label >)/
(TEST1 <fonction grammaticale > <label >)
<action > :: = < label >* (SETRI <fonction grammaticale >
< valeur >)/
< label >* (SETRI <catégorie grammaticale >
< valeur >)/
< label >* (SETR2 < fonction grammaticale >
< entier >)/
(MOUV <zone > * <valeur > *)/
(CONS < catégorie grammaticale >).
<sortie> :: = (NEXT < entier >)/
(TO <label >)
< nom d'état > :: = TR. <catégorie grammaticale >
<label > :: = <catégorie grammaticale >. T
< fonction grammaticale > :: = SUJ/PRED/CPN/COD/CPV/CPX
<catégorie grammaticale > :: = SN/SVER/SPP/RQUE/RQUI/INV/
NEG/Q1/Q2/Q3/FIN
<valeur > :: = */NIL/NEG/POS/DCL/INT
<entier > :: = 0/1/2 . . .

La première ligne signifie que le réseau élargi est représenté par une parenthèse gauche suivie par un ensemble d'arcs qui est lui-même suivi d'un nombre quelconque d'ensemble d'arcs (zéro ou plus) et d'une parenthèse droite. Un ensemble d'arcs est un nom d'état suivi d'un nombre quelconque d'arcs, etc...

L'arc CAT peut être suivi si la catégorie grammaticale qu'on lit est la même que celle qui figure après CAT.

Ainsi (TRSN (CAT SN) ... ne peut être suivi que quand on lit un syntagme nominal SN.

Les actions SETR1 permettent de construire les fonctions grammaticales telles qu'elles apparaissent dans la structure profonde de la phrase :

SUJ - sujet, PRED - prédicat, CPN - complément nominal, CPV- complément verbal, COD - complément d'objet direct, CPX - complément indéfini.

Le complément indéfini CPX est un syntagme prépositionnel qui suit un complément d'objet direct.

(SETR1 SUJ *)

signifie qu'il faut garnir un registre correspondant à SUJ, avec la valeur entière qui est le rang du syntagme nominal qu'on vient de lire.

(SETR1 COD NIL)

signifie qu'il faut garnir le registre correspondant à COD avec la valeur nulle.

SETR1 permet aussi de construire des catégories grammaticales. Ceci pour un développement ultérieur qui utiliserait le réseau élargi pour lever des ambiguïtés.

Les conditions TEST1 permettent de tester la présence ou l'absence de certaines catégories ou fonctions grammaticales. Le label qui figure dans une condition indique l'action qu'on doit réaliser si la catégorie testée est absente :

(TEST1 VER CPNT)

revient à tester si le syntagme verbal est déjà lu. Si non faire l'action CPNT.

(MOUV SIGNE "NEG") garnit la zone SIGNE avec la valeur "NEG"; (CONS SN) permet de poursuivre la construction de la structure partielle de la phrase avec un syntagme nominal placé comme numérateur de la catégorie qui vient d'être lue

La sortie indique l'acte final dans un arc. (NEXT n) signifie qu'il faut lire la nè catégorie suivante :
(TO 'label) indique l'action suivante à réaliser :

(NEXT 1) lire la catégorie suivante
(NEXT 2) lire la 2ème catégorie qui suit
(TO SNT) réaliser les actions SNT.

Les noms d'états du réseau sont donnés sous la forme de TR' concaténé avec la catégorie grammaticale correspondant à l'arc.

TRSN pour l'arc (CAT SN) ...
TRSPP pour l'arc (CAT SPP), ...

Les labels SNT, CODT, CPVT, ... servent à referencer un certain nombre d'actions.

Le réseau grammatical elargi est donné en annexe. Nous en reprenons ici une partie pour illustrer son fonctionnement (les arcs sont numérotés pour la commodité).

0. (S_o)
1. (TRSN (CAT SN)
(TEST1 VER SNT) (TEST1 INV CODT)
SENT (SETR1 S *) (SETR1 SUJ *)
(CONS S) (NEXT 1)
CODT (SETR1 COD *) (NEXT 1)
SNT (SETR1 SN *) (CONS SN) (NEXT 1))
2. (TRSPP (CAT SPP)
(TEST1 VER CPNT) (TEST1 COD CPVT)
CPXT (SETR1 CPX *) (NEXT 1)
CPVT (SETR1 CPV *) (NEXT 1)
CPNT (SETR1 CPN *) (NEXT 1)

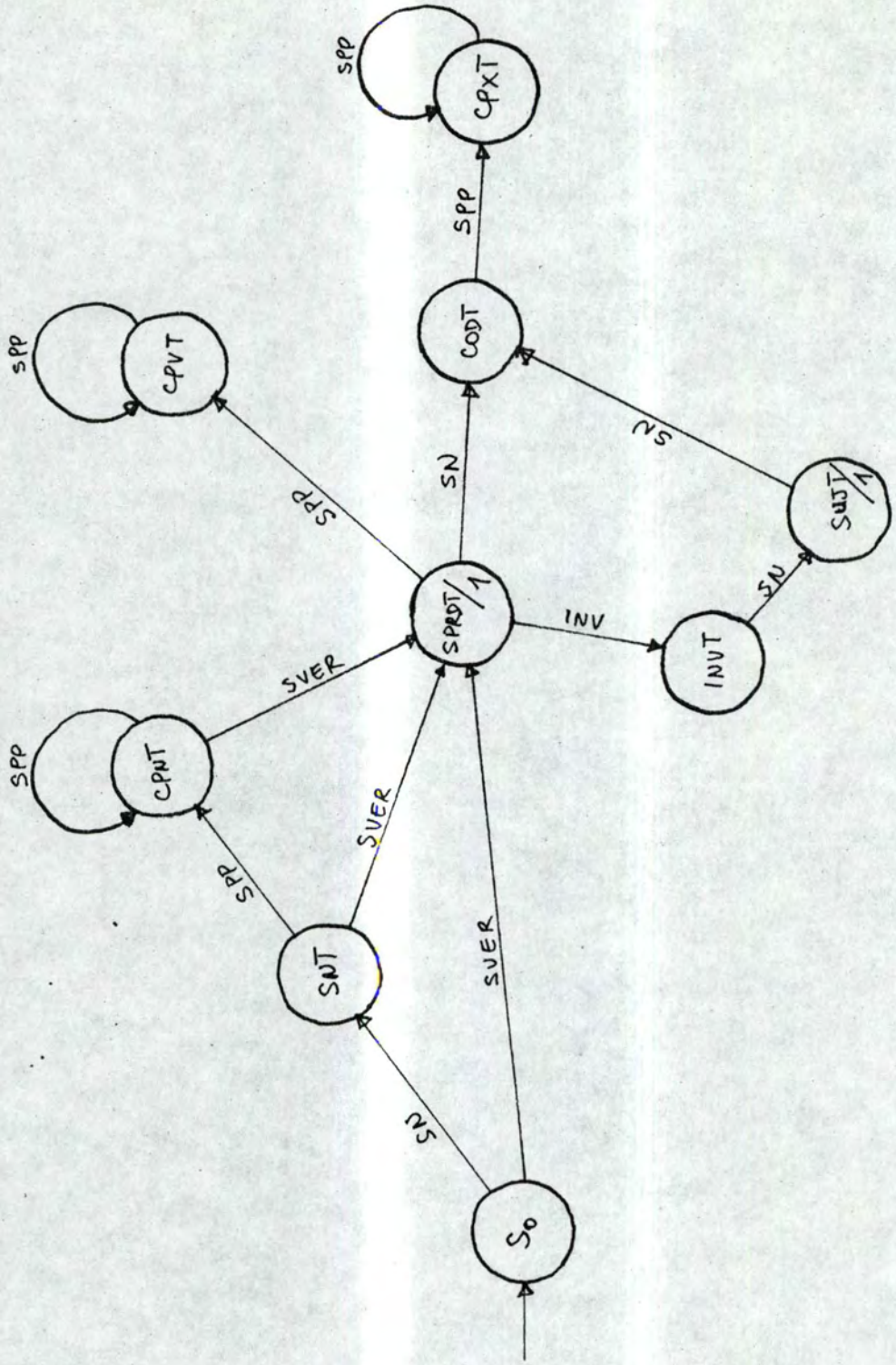


fig. 8.- Représentation partielle du réseau grammatical élargi

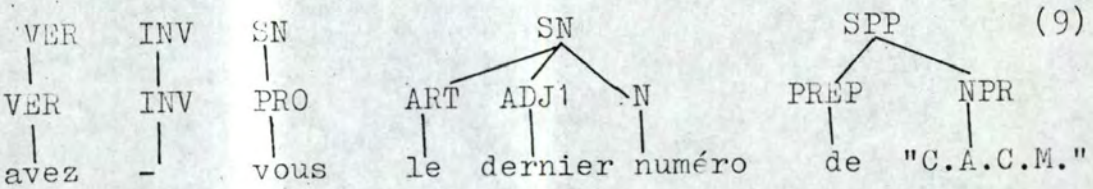
```
3. (TRVER (CAT   VER)
          (TEST1  SN  SPRDT)
     SUJT (SETR2  SUJ  0) (CONS  S) (SETR2  S  0)
     SPRDT (SETR2  S  0) (SETR1 SVER  *)
          (CONS  SVER) (NEXT 1))
4. (TRINV (CAT  INV)
          (TEST1  SUJ  INVT)
          (MOUV  TYPE  "INT") (NEXT 2)
     INVT (MOUV  TYPE  "INT") (NEXT 1))
```

S_o est l'état initial du réseau. Le diagramme (8) donne une représentation graphique de cette partie de réseau.

Reprenons la phrase :

"avez-vous le dernier numéro de "C.A.C.M." ?

dont nous rappellons la structure partielle :



L'analyse de cette structure se fait comme suit :

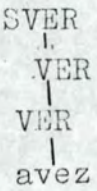
1) la lecture de VER permet la transition vers l'arc 3. Et puisqu'aucun syntagme nominal n'est encore lu,

(TEST1 SN SPRDT)

fait réaliser les actions SPRDT :

(SETRI SVER *) - garnir le registre de SVER avec 1

(CONS SVER) - compléter la structure partielle de la façon suivante :

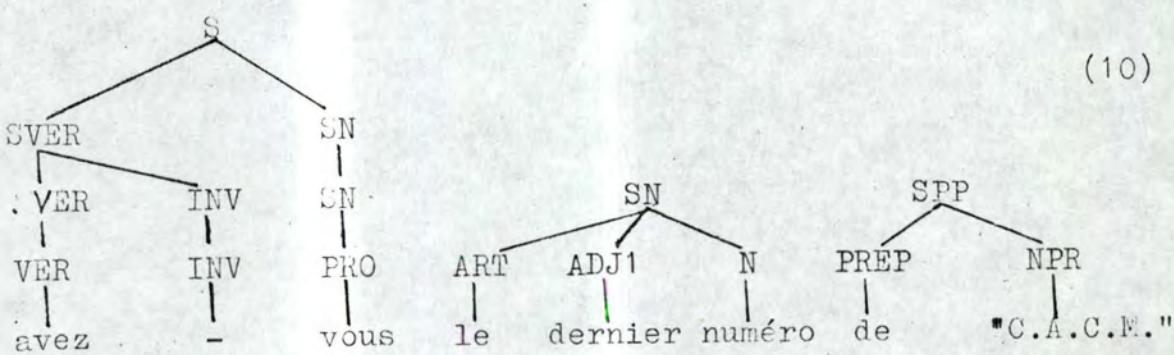


(NEXT 1) : lire la catégorie suivante.

2) INV permet la transition vers 4 :
(TEST1 SUJ INVT) - fait réaliser les actions INVT puisqu'il n'y a pas encore de sujet SUJ.
(MOUV TYPE "INT") (NEXT 1). On garnit la zone TYPE avec "NEG" et on lit la catégorie suivante. Le registre INV est remis à zéro.

3) SN met le réseau dans l'état 1. Ces deux conditions qui suivent : (TEST1 VER SNT) (TEST1 INV CODT) font réaliser les actions SENT : "vous" est pris comme sujet et la phrase acceptée. On lit le symbole suivant :

A ce niveau, la structure de la phrase est: (10)



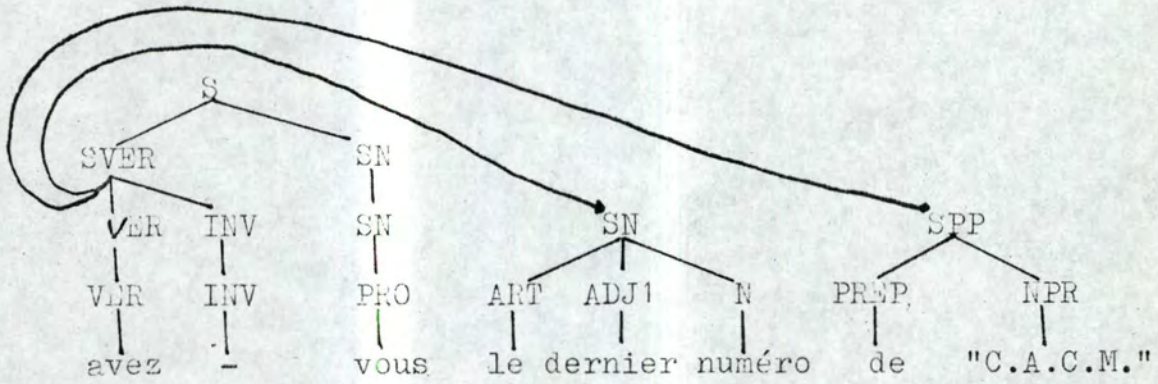
4) le deuxième SN ramène le réseau à l'état 1.
(TEST1 INV CODT) fait réaliser les actions CODT du fait que le registre INV est à zéro. Le syntagme nominal "le dernier numéro" est pris comme complément d'objet direct et le registre COD est garni avec la valeur 4.

5) La lecture du syntagme prépositionnel SPP permet la transition avec l'arc. 2. Et puisque les deux registres VER, COD sont non nuls, on réalise les actions CPXT :

(SETR1 CPX *) - le syntagme "de "C.A.C.M."" est pris comme un complément indéfini.

D'où la structure finale de la phrase :

(11)



6.4. Algorithme d'analyse syntaxique

L'analyse syntaxique se fait donc en deux étapes essentielles :

a) en utilisant la grammaire G_1, on construit une structure partielle de la phrase

b) le réseau grammatical élargi, part de cette structure partielle pour dégager la structure de surface entière de la phrase tout en indiquant les fonctions grammaticales de la structure profonde de la phrase.

D'où l'algorithme suivant ASX :

Soit $x_1, x_2, \ldots x_m$ une chaîne de mots de V

ASX 1. La consultation du dictionnaire trouve les catégories grammaticales des x_i :

$$c_i = f(x_i)$$

d'où la chaîne :

$$C_1 \quad C_2 \quad \ldots \quad C_m \qquad (12)$$

ASX 2. Déterminer les intervalles dans (1.2.) en consultant la table de réduction (5.3.)

ASX 3. Construire la structure partielle en appliquant la règle de réduction R à la suite (12).

ASX 4. Appliquer à la structure partielle le réseau élargi.

6.5. Sortie de l'analyseur syntaxique

A l'issue de l'analyse syntaxique, la structure de la phrase est décrite de façon complète par :

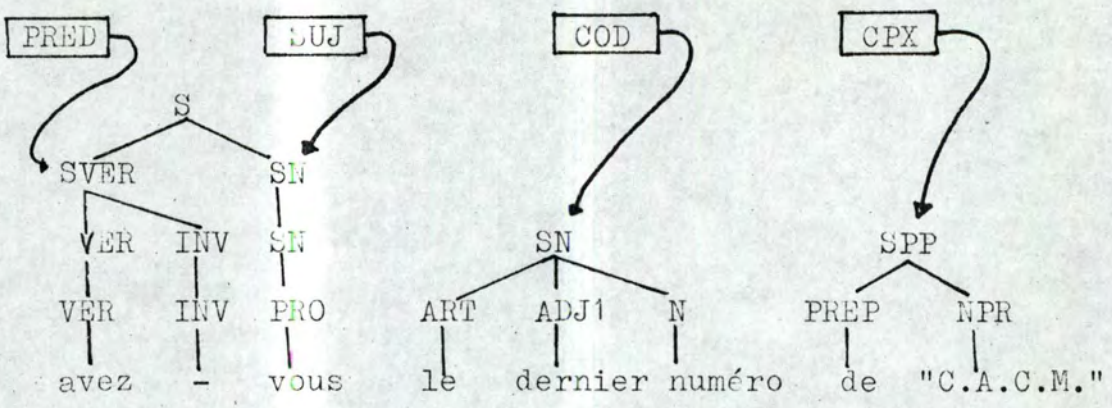
a) un arbre syntaxique - qui est la structure de surface de la phrase

b) deux zones FORME et SIGNE qui indiquent respectivement la forme (déclarative ou interrogative) et le signe (positif ou négatif) de la phrase. Ces deux zones sont remplies par le réseau élargi à travers l'action MOUV.

c) un ensemble de registres correspondant aux différentes fonctions grammaticales possibles : SUJ, PRED, CPN, CPV, COD, CPX et qui pointent vers les noeuds correspondant de l'arbre syntaxique. Ces fonctions grammaticales sont celles de la structure profonde de la phrase.

Ainsi, pour la phrase :
"avez-vous le dernier numéro de "C.A.C.M." la sortie sera

FORME INT
SIGNE POS



Pour ne pas surcharger le dessin, nous n'avons pas tracé les liaisons du syntagme verbal (SVER) avec les deux compléments SN (COD) et SPP (CPX).

Le registre PRED précède toujours le registre SUJ et les registres compléments COD, CPV, CPX. Ce qui fournit une forme de notation polonaise des fonctions grammaticales :

PRED	avez
SUJ	vous
COD	le dernier numéro
CPX	de "C.A.C.M."

Pour simplifier l'arbre syntaxique, nous ne construirons pas le noeud SPRD qui n'apporte aucune information supplémentaire.

7. INTERPRETATION SEMANTIQUE

7.1. Modèle de base

Nous reprenons ici la définition de Bobrow telle qu'elle a été développée par Woods (3.1.). Le modèle sémantique du système d'interrogation est un quintuplet :

$$\{O, F, R, P, S\}$$

1) $O = \{O_i\}$ est l'ensemble des objets de B referencés par des syntagmes nominaux :

- un titre
- le titre de l'article de Woods

Pour illustrer le fonctionnement de l'interpréteur, nous supposerons que ces objets sont représentés dans le tableau suivant :

.AUTEUR	TITRE	COLLECTION	EDITEUR	ANNEE	COTE	FORMAT
AUT1	TITR1	COL1	EDIT1	AN1	COT1	FORM1
AUT1	TITR2	COL2	EDIT2	AN2	COT2	FORM2
AUT2	TITR3	COL1	EDIT2	AN2	COT3	FORM3
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure_2

Chaque colonne du tableau reprend une classe d'objets de B :

AUT1, AUT2, ... sont de la classe AUTEUR

TITR1, TITR2, ..., de la classe TITRE.

2) $F = \{F_i\}$ est l'ensemble des applications F_i de la forme :

l'auteur de ———
la côte de ———

Ainsi, "l'auteur de TITR1" est la fonction qui fait correspondre à l'objet TITR1, l'objet AUT1. De même "la côte de TITR3" fait correspondre à TITR3 COT3.

Un objet obtenu à l'issue de l'application d'une fonction F_i est dit fonctionnement déterminé :
AUT1, résultat de "l'auteur de TITR1" est un objet fonctionnellement déterminé.

3) $R = \{R_i\}$ est l'ensemble des relations définies sur O.
Ce sont d'abord les relations traduites par une ligne du tableau précédent.

——— est l'auteur de ———
——— est la collection de ———
——— est le format de ———

Il y a ensuite les relations qui indiquent certaines propriétés des objets O :

vous avez ———
——— est disponible

Nous supposerons que toutes ces relations sont représentées dans B.

4) $P = \{P_i\}$ - qui sont les réalisations de R_i sur des arguments, objets de O :

AUT2 est l'auteur de TITR3
vous avez la revue TITR5
le livre TITR1 est disponible.

5) L est l'ensemble des règles sémantiques qui permettent de déduire des nouvelles propositions et qui sont définies ci-après.

7.2. Définition des primitives sémantiques

Nous distinguerons deux catégories de primitives sémantiques :

- les prédicats qui permettent d'énoncer des propriétés des objets X, propriétés que nous supposons indiquées dans B.
- les fonctions qui appliquées à des arguments (objets X) produisent d'autres objets X.

7.2.1. Les prédicats

a) Un premier groupe de prédicats correspond aux données élémentaires de la figure 2 et portent sur la classe des X.

AUTEUR (X)	-	X est un auteur
TITRE (X)	-	X est un titre
COLLECTION (X)	-	X est une collection
EDITEUR (X)	-	X est un éditeur
ANNEE (X)	-	X est une année d'édition
COTE (X)	-	X est une côte
FORMAT (X)	-	X est un format

Pour un objet (X), TITRE (X), quatre autres prédicats précisent la nature de X, nature que nous supposons marquée dans B comme propriété de ces objets X :

LIVRE(X)	-	X est un livre
REVUE(X)	-	X est une revue
JOURNAL(X)	-	X est un journal
ARTICLE (X)	-	X est un article

b) Les prédicats qui correspondent aux opérateurs logiques :

ET (P_1, P_2) - P_1 et P_2
OU (P_1, P_2) - P_1 ou P_2
NON (P) - la négation de P
EQ (X, Y) - X est égal à Y

où P, P_1, P_2 sont des propositions, X, Y - des objets.

c) un prédicat d'existence pour X, tel que TITRE (X) :

POSS (X)

signifie que l'objet X existe dans B. Le prédicat

DISPO (X)

est vrai si l'objet X est disponible. Ces prédicats correspondent respectivement aux syntagmes verbaux : " ——— avez ———", " ——— est disponible", " ——— est libre"!

7.2.2. Les fonctions

a) AUT (X) - l'auteur de X
TITR (X) - le titre de X
COLL (X) - la collection de X
EDIT (X) - l'éditeur de X
ANN (X) - l'année d'édition de X
COT (X) - la côte de X
FORM (X) - le format de X

b) la fonction qui indique la classe de l'objet X :

CLASS (X)

Et si CLASS (X) = TITRE, une fonction

NAT (X)

permet de préciser s'il s'agit respectivement d'un livre, d'une revue, d'un article ou d'un journal.

c) si CLASS (X) = AUTEUR, une fonction

OUVR (X)

indique une oeuvre (livre, article, revue, journal) de X.

7.3. Définition du langage de commande

En vue de permettre l'interprétation de phrases déclaratives destinées à faire une mise à jour de B, nous introduisons en plus des instructions de base LIST (X) et TEST (P) définis en (3.3), l'instruction de base

INFO (P)

dont le but est de mettre en oeuvre des procédures qui traduisent la véracité de P. Ainsi la commande :

INFO (DISPO (TITR1))

servira à marquer de façon appropriée dans B que TITR1 est disponible. INFO (P) correspond à l'interprétation des phrases déclaratives (affirmatives ou négatives).

Pour achever la définition du langage de commande, il faut maintenant adapter les quantificateurs :

(FOR QUANT X / CLASSE: R (X) : P (X))

au français, en redéfinissant QUANT. Nous réécrirons ces quantificateurs sous la forme suivante :

(POUR QUANT X / CLASSE : R (X); P(X))

Les différentes valeurs de QUANT et leur choix seront précisées dans les D-règles. Les quantificateurs numériques :

a) n objets X vérifient les propositions P(X) et R(X)

(POUR n PLUS X / CLASSE: R (X); P (X))

b) au moins n objets X vérifient P (X) et R (X)

(POUR SUP (n) PLUS X / CLASSE: R (X); P(X))

c) au plus n objets X vérifient P (X) et R (X)

(POUR INF (n) PLUS X / CLASSE: R (X); P (X))

ne seront mentionnés ici que pour mémoire : nous ne nous occupons pas de leur génération dans ce travail.

7.4. Les règles sémantiques

Dans la sémantique de Woods, les règles sémantiques sont basées sur la reconnaissance de sous-arbres de la forme de ceux présentés en (3.5.) :

G_1 : sujet-verbe

G_2 : verbe-complément d'objet direct

G_3 : verbe-complément verbal

L'analyseur syntaxique défini ci-dessus (6.4.) ayant déjà dégagé les fonctions grammaticales, cette reconnaissance n'est plus nécessaire. Il en résulte une grande simplification des règles sémantiques dont les conditions ne portent plus que sur le type de la phrase et les mots de la phrase (ou noeuds terminaux de l'arbre syntaxique).

Considérons par exemple la phrase :

Le livre "Automated Language Processing" est disponible (1)

Outre l'arbre syntaxique, l'analyseur donne les informations suivantes :

FORME = DCL

SIGNE = POS (2)

PRED = EST DISPONIBLE

SUJ = AUTOMATED LANGUAGE PROCESSING

(1) étant une phrase déclarative dont le but est de marquer que "AUTOMATED LANGUAGE PROCESSING" est disponible. Cette action est traduite dans le langage de commande par

INFO (DISPO (———)) (2')

La règle sémantique aura donc pour rôle d'associer (2) à (2') qui sera considéré comme l'interprétation de (1).

Pour reconnaître la primitive DISPO, il faut vérifier que les conditions qui définissent DISPO sont réalisées :

- le syntagme verbal est "est disponible" ou "est libre"
- la classe du sujet est TITRE

La S-règle s'écrirait donc :

FORME = dcl
SIGNE = pos
PRED. VER = être
PRED. ADJ2 = disponible, libre
CLASS (SUJ.N) = TITRE

⟹ INFO (DISPO (SUJ)) (3)

et qui s'interprête de la façon suivante :

a) si la zone FORME contient la valeur "dcl"
b) si la zone SIGNE contient la valeur "pos"
c) si le verbe du prédicat (PRED.VER) est le verbe "être" conjugué au présent de l'indicatif
d) si le prédicat contient un adjectif de la catégorie ADJ2 qui est soit "disponible", soit "libre"
e) si le nom contenu dans le syntagme nominal sujet (SUJ.N) désigne un objet de la classe TITRE, alors l'interprétation de la phrase est :

INFO (DISPO (SUJ))

où SUJ doit être remplacé par sa valeur ici

AUTOMATED LANGUAGE PROCESSING

pour obtenir le résultat final :

INFO (DISPO (AUTOMATED LANGUAGE PROCESSING))

L'ensemble des règles sémantiques qui sert ici d'illustration est donné en Annexe A.4. Les parties droites de ces règles ont la même syntaxe que dans les règles de Woods. Lorsque des conditions de la partie gauche portent sur des constructions répétées, celles-ci sont numérotées dans l'ordre où elles apparaissent à la sortie de l'analyseur syntaxique.

7.4.1. S-règles

Le but essentiel de ces règles est d'indiquer l'instruction de base et les primitives sémantiques qui correspondent à l'interprétation du prédicat (PRED), de la forme (FORME) et du signe (SIGNE) de la phrase tels que ceux-ci sont dégagés à l'issue de l'analyse syntaxique.

a) interprétation de FORME

A l'exception de S7 qui représente les phrases impératives, à la valeur "DCL" de FORME, correspond l'instruction de base INFO qui nous le rappelons est considérée comme une instruction de mise à jour de B.

A l'inverse, la valeur "INT" appelle les instructions de base TEST (S2, S6) ou LIST (S4).

b) interprétation de SIGNE

Lorsque SIGNE = NEG, on ajoute le prédicat NON à l'interprétation du prédicat PRED de la phrase.

Exemple : vous avez ———

vous n'avez pas ———

donnent respectivement :

INFO (POSS (COD))

INFO (NON (POSS (COD)))

Dans les sous-règles présentées en A.4., nous ne présentons que les cas où SIGNE = pos

c) interprétation de PRED

Cette interprétation fournit les noms de primitives sémantiques qui interviennent comme arguments des instructions de bases INFO, TEST et LIST. Nous donnons dans une table S-TABLE l'ensemble des s-règles qui correspondent à chaque verbe. Pour éviter les interprétations redoutantes (lorsque dans une phrase donnée plusieurs s-règles sont valides), les s-règles seront ordonnées de façon adéquate dans S-TABLE de façon que celles qui imposent moins de conditions sur la structure de la phrase suivent celles qui en imposent plus.

VERBES	S-REGLES
AVOIR	S1, S2
ETRE	S6, S5, S4, S3
DONNER	S7
ECRIRE	S7

Fig. 5.- S-TABLE

7.4.2. D-règles

Elles servent à définir, QUANT dans les quantificateurs

(POUR QUANT X / CLASSE : R (X); P (X))

a) les articles définis LE, LA, LES sont directement repris comme valeurs de QUANT

b) lorsqu'un nom est précédé d'un article indéfini UN, UNE ou de l'adjectif QUELQUES, QUANT a la valeur "PLUS" (mis pour plusieurs).

c) QUANT est égal à "TOUT" si le nom est précédé de tout ou chaque.

Exemples

Les syntagmes nominaux :

un livre de AUT1
les livres de AUT1
chaque livre de AUT1

sont respectivement quantifiés de la façon suivante :

(POUR PLUS X / ∇ ; Δ)
(POUR LES X / ∇ ; Δ)
(POUR TOUT X / ∇ ; Δ)

Les D-règles sont données dans une table D-TABLE.

DETERMINANT	D-REGLE
UN, UNE	D1
QUELQUES	D1
LE, LA, L	D2
LES	D3
TOUT	D4
CHAQUE	D4

fig. 4.- D-TABLE

7.4.3. N-règles

Elles servent à déterminer le domaine à quantifier (CLASSE) dans un syntagme nominal. Pour illustrer les problèmes qui se posent ici considérons la question suivante :

Qui est l'auteur de TITR1 ?

Quelle que soit l'implémentation des primitives qui interviennent dans l'interprétation de cette phrase, on voudrait pouvoir pointer directement vers TITR1 pour trouver son auteur et non examiner tous les auteurs possibles pour voir qui est l'auteur de TITR1.

En d'autres termes, l'interprétation :

(POUR LE X / AUT (TITR1) : ———; LIST (X))

est surement préférable à :

(POUR LE X / AUTEUR : EQ (X, AUT (TITR1)); LIST (X))

Cela revient à dire que lorsqu'il s'agit d'un objet fonctionnellement déterminé, il faut ramener le domaine à quantifier à l'objet lui-même.

Les règles N1 à N5 donnent les domaines correspondant aux noms simples : auteur, année, ...

Les règles N6 à N10 correspondent aux noms d'objets fonctionnellement déterminés : l'auteur de ———, le livre de ———, ...

La table N-TABLE sera ordonnée de façon à ce que les règles correspondant aux noms des objets fonctionnellement déterminés précèdent celles des noms simples.

NOMS	N-règles
livre	N6, N1
revue	N6, N1
article	N6, N1
ouvrage	N6, N1
auteur	N7, N2
année	N8, N3
côte	N9, N4
format	N10, N5
collection	N11, N12

fig. 5.- N-TABLE

7.4.4. R-règles

Elles permettent de dégager les restrictions R (X) aux domaines à quantifier par l'interprétation des adjectifs, de certains compléments nominaux et des appositions.

Les règles R1, traitent des appositions de la forme :

le livre "Automated Language Proc."

la collection "U"

Les règles R4 à R7 permettent de préciser la classe TITRE par un des prédicats LIVRE (X), REVUE (X), ARTICLE (X), JOURNAL (X).

A la différence des autres tables, la table des R-règles, R-TABLE fournit pour un nom donné des règles qui ne sont pas exclusives mais plutôt à combiner pour trouver la bonne restriction R (X) au domaine à quantifier.

Chaque règle sert à affiner cette restriction.

Par exemple, les règles R5 et R1 appliquées à :

la revue "C.A.C.M."

produisent respectivement les restrictions suivantes :

REVUE (X)

EQ (TITR (X), C.A.C.M.)

qui permettent de générer le quantificateur suivant :

(POUR LE X / TITRE : (REVUE (X), EQ (TITR (X), C.A.C.M.); Δ)

NOMS	R-règles
livre	R4, R1, R2
revue	R5, R1, R8
article	R6, R1, R2
journal	R7, R1
collection	R2
titre	R3, R8

Fig. 6.- R-TABLE

7.5. Interprétation sémantique

La logique est la même que celle de l'interpréteur de Woods (3.8). Nous continuerons à distinguer un S-processeur qui analyse le prédicat PRED et un NP-processeur qui analyse les autres fonctions grammaticales. Il y a quatre phrases essentielles dans l'interprétation d'une phrase :

- l'interprétation du prédicat par le S-processeur suivie de trois phrases qui correspondent à l'évaluation d'une quelconque autre fonction grammaticale par le

NP-processeur

- détermination du quantificateur avec les D-règles
- détermination du domaine à quantifier avec les N-règles
- détermination des restrictions du domaine à quantifier avec les R-règles.

On détermine l'interprétation en remplaçant les arguments par les évaluations ainsi établies par le NP-processeur.

Considérons l'exemple suivant :

Qui est l'auteur de TITR1 ?

L'analyseur syntaxique produit le résultat suivant :

FORME = int
SIGNE = pos
PRED = est
SUJ = l'auteur (4)
CPN = de TITR1
COD = qui

1) Le S-processeur commence par reconnaître le prédicat

PRED = est

et lit dans S-TABLE, les sous-règles figurant dans l'entrée du verbe "être" : S6, S5, S4, S3. Le S-processeur essaie les différentes S-règles dans l'ordre où elles sont lues pour retenir la première dont les conditions sont vérifiées par (4)

La règle S4 :

FORME = int
SIGNE = pos ⟹ LIST (SUJ) (5)
PRED = être
COD = qui, quel

est vérifiée. Le S-processeur crée une zone de travail W qu'il initialise avec la partie droite de la règle S4 :

W : = LIST (SUJ) (6)

Le S-processeur appelle alors le NP-processeur pour évaluer le syntagme nominal SUJ.

2) A chaque fois qu'il est sollicité, le NP-processeur commence par créer une nouvelle variable en concaténant X à un compteur qui sert à distinguer les différents syntagmes nominaux :

X1 pour le 1er
X2 pour le 2ème etc...

3) La première tâche que réalise le NP-processeur est la détermination du quantificateur de SUJ. Pour cela il lit l'article qui accompagne le nom qui figure dans SUJ et cherche dans le D-TABLE, la D-règle correspondant à cet article : ici l'article est l (mis pour le) dont la D-règle est D2 :

SN. ART = le, la, l ⟹ (POUR LE X / ∇ ; Δ)

Le NP-processeur crée une zone de travail Z dans laquelle il met la partie droite de D2 :

Z : = (POUR LE X1 / ∇ ; Δ) (7)

4) La deuxième tâche du NP-processeur est la détermination de la classe de l'objet referencé par le syntagme nominal. L'entrée du nom "auteur" dans la table N-TABLE indique les règles N7, N2. La règle N7 :

SN.N = auteur
CLASS (CPN.N) = TITRE ⟹ AUT (CPN) : ∇ (8)
CPN.PREP = de, du, des

est vérifiée. Le NP-processeur remplace alors dans (7) le symbole par la partie droite de (8) :

Z : = (POUR LE X1 / AUT (CPN) : ∇ ; Δ) (9)

Rappelons que la règle (8) signifie que SUJ referencie un objet fonctionnellement déterminé

auteur de ——

5) Il reste à déterminer les éventuelles restrictions sur AUT (CPN). Le nom "auteur" n'ayant pas d'entrée dans R-TABLE, les restrictions n'existent pas.

Le symbole dans (9), qui devait recevoir ces restrictions est remplacé par un trait-d'union (par le NR-processeur). Et (9) devient :

Z : = (POUR LE X1 / AUT (CPN) : - ; Δ) (10)

6) Le NP-processeur ayant terminé l'évaluation directe de SUJ, retourne la valeur Z au S-processeur. Le S-processeur appelle à nouveau le NP-processeur pour évaluer l'argument CPN de la primitive AUT. Le nom qui figure dans CPN étant un nom propre "TITR1", le NP-processeur renvoie la même valeur au S-processeur que le place dans (10) :

Z : = (POUR LE X1 / AUT (TITR1) : - ; Δ) (11)

7) Z qui est l'évaluation finale de SUJ contient un symbole (contrairement à la valeur "TITR1" de CPN ci-dessus). Le S-processeur remplace alors le symbole par (6) pour produire :

Z : = (POUR LE X1 / AUT (TITR1) : - ; LIST (X1))

qui est l'interprétation finale de la phrase.

D'où l'algorithme d'interprétation sémantique (ASM) :

ASM S1 - Le S-processeur détermine le verbe de la phrase donné dans PRED. En consultant dans la table S-TABLE, l'entrée correspondant à ce verbe, le S-processeur détermine la S-règle qui décrit la structure de la phrase

ASM S2 - Le S-processeur met dans une zone detravail W la partie droite de cette S-règle et appelle le NP-processeur pour interpréter les arguments des différentes primitives dans W.

ASM N0 - Si le syntagme nominal de l'argument se réduit à un nom propre, celui-ci est renvoyé par le NP-processeur comme valeur de l'argument. Le S-processeur place cette valeur dans W à la place de l'argument.

ASM N1 - Dans le cas où le syntagme nominal ne se réduit pas à un nom propre, le NP-processeur génère un quantificateur :

(POUR QUANT X1 / CLASSE : ∇ ; Δ)

où X1 est une nouvelle variable associée au syntagme nominal. Ce quantificateur est placé dans une zone Z.

ASM N2 - Le NP-processeur détermine QUANT au moyen des D-règles figurant dans D-TABLE.

ASM N3 - Le NP-processeur détermine CLASSE au moyen des N-règles de N-TABLE

ASM N4 - Le NP-processeur détermine enfin les restrictions au moyen des R-règles de R-TABLE et les met à la place du symbole ∇ dans Z.

ASM S3 - Le S-processeur remplace alors dans Z, Δ par W.

7.5. Exemples

1.- TITR1 est disponible

```
INFO (DISPO (TITR1))
```

2.- le livre TITR1 est disponible

```
(POUR LE X1 / TITRE : (LIVRE (X1), EQ (TITR (X1),
 TITR1)); INFO (DISPO (X1)))
```

3.- Qui est l'auteur de TITR1 ?

```
(POUR LE X1 / AUT (TITR1) : - ; LIST (X1))
```

4.- AUT1 est-il l'auteur de TITR1 ?

```
(POUR LE X1 / AUT (TITR1) : - ; TEST (EQ (AUT1, X1))
```

5.- TITR1 est-il un livre de AUT1 ?

```
(POUR PLUS X1 / OUVR (AUT1) : LIVRE (X1)
                TEST (EQ (TITR1, X1)))
```

6.- Vous avez le livre TITR1

```
(POUR LE X1 / TITRE : (LIVRE (X1), EQ (TITR (X1),
 TITR1)); INFO (POSS (X1)))
```

7.- Avez vous la revue TITR3 ?

```
(POUR LE X1 / TITRE : (REVUE (X1), EQ (TITR (X1),
 TITR3)); TEST (POSS (X1)))
```

8. CONCLUSION

8.1. Résumé

Dans ce travail, nous nous sommes attachés à présenter des méthodes uniformes d'analyse syntaxique et d'interprétation sémantique des phrases d'un langage naturel d'interrogation.

Nous avons d'abord défini un langage naturel dont les phrases ont la forme qu'elles ont habituellement en français Un analyseur syntaxique basé sur une grammaire catégorielle et un réseau grammatical élargi construit la structure syntaxique des phrases (arbre syntaxique), en dégage le type (positive/négative, affirmative/interrogative) et les fonctions grammaticales (sujet, prédicat, compléments d'objets directs, compléments verbaux, compléments nominaux).

En utilisant des règles sémantiques, le S-processeur (au moyen des S-règles) et le NP-processeur (au moyen des D-règles, N-règles et R-règles) traduisent les informations obtenues de l'analyseur syntaxique en une expression du langage de commande, expression qui représente le 'sens' de la phrase.

Bien que ces techniques ne sont pas complètement générales pour s'appliquer à toute une langue naturelle, elles restent néanmoins valables dans un grand nombre de cas où il est fait appel à un ordinateur pour comprendre une phrase en langage naturel et prendre une action appropriée : gestion de stocks, documentation, gestion hospitalière, gestion de compagnie de transports, etc...

8.2. Puissance des techniques

On peut maintenant se demander quelle est la puissance des techniques proposées pour répondre à une question ? D'un point de vue théorique, le langage de commande est en gros équivalent à un calcul de prédicat du premier ordre (à l'exception des instructions de base LIST et INFO qu'on ne peut pas interpréter par un calcul de fonctions de vérité). La puissance logique des techniques est donc considérable.

D'un point de vue pratique, le système permet de poser des questions très diverses sur la base de données B. Quatre types de phrases interrogatives :

- l'inversion simple
- les phrases introduites par qui
- les phrases introduites par quel, quelle, quels
- les phrases commençant par est-ce que

sont permises. Il permet de faire des mises-à-jour et des éditions (par les deux ordres 'donnez' et 'écrivez'). Le système analyse les phrases relatives introduites par 'que' et 'qui' ainsi que les adjectifs et compléments nominaux qui modifient un nom. Les appositions et noms fonctionnellement déterminés sont étudiés.

8.3. Extension

L'analyseur syntaxique tel qu'il est implémenté peut être amélioré afin de pouvoir affiner son analyse et traiter de cas qui ne sont pas étudiés ici : le problème étant de faire un bon arbitrage entre le coût de toute complication de l'analyseur et le gain réel d'informations que l'on obtient.

On pourra par exemple étudier :

- les connecteurs 'et', 'où', ',', ...
- les copules et modalités
- d'autres temps que le présent
- des phrases qui ont un nombre quelconque de relatives
- la forme passive, etc...

L'extension du langage de commande se fait aisément en ajoutant de nouvelles primitives, prédicats, fonctions et instructions de bases comprises comme des sous-routines que la composante de recherche peut appeler. On introduira en plus de nouvelles règles sémantiques correspondant aux nouveaux concepts. Si, par exemple, on veut introduire dans le catalogue qui nous sert d'exemple la classe SUJET qui donne pour chaque objet TITRE, son sujet, il faudrait ajouter les primitives suivantes :

SUJET (X) - X est un sujet
SUJ (X) - le sujet de X

Les nouvelles règles sémantiques ont une N-règle correspondant au nom 'sujet' et des R-règles pour interpréter les expressions 'relatif à' et 'portant sur' par exemple.

8.4. Comparaisons

Par rapport aux travaux de Woods dont ce mémoire s'inspire en grande partie, nous faisons ici une synthèse entre les théories développées dans sa thèse [34] où il ne s'occume que du problème de l'interprétation sémantique et sa notion récente de réseau grammatical élargi [36] .

L'utilisation d'une grammaire catégorielle de reconnaissance qui construit les constituants immédiats (syntagmes nominaux, syntagmes verbaux, syntagmes prépositionnels) simplifie le réseau grammatical élargi. Par ailleurs, la détermination des fonctions grammaticales allège considérablement l'interprétation sémantique et les règles sémantiques.

8.5. Point de vue de l'utilisateur

Un système d'interrogation en langage naturel offre, sans aucun doute, beaucoup de facilités à l'utilisateur qui n'est pas obligé de passer par un langage de programmation spécialisé pour communiquer ou obtenir des informations de l'ordinateur.

Mais si, nous considérons les deux phrases :

TITR1 est disponible
Le livre TITR1 est disponible

et leurs interprétations respectives :

INFO (DISPO (TITR1))

(POUR LE X1/TITRE : (LIVRE (X1), EQ (TITR (X1), TITR1));
INFO (DISPO (X1)))

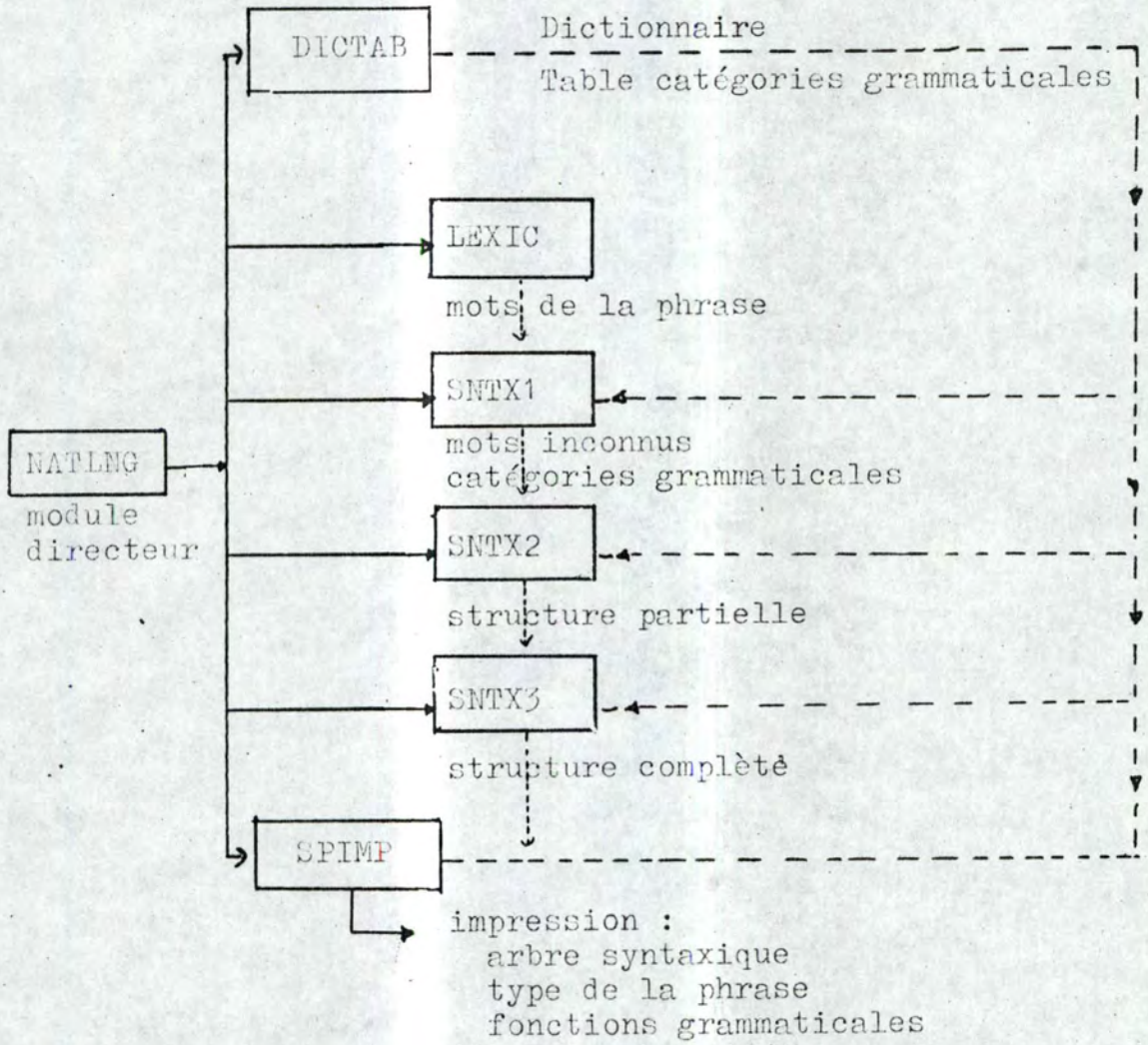
nous constatons la répercussion (et le coût) de la précision apportée dans la deuxième phrase en indiquant que TITR1 est un livre. L'utilisateur aurait donc intérêt à bien connaître toute la logique de l'interprétation sémantique afin d'en tirer parti.

III. MISE EN OEUVRE

9.- ANALYSEUR SYNTAXIQUE

9.1. Présentation Générale

L'analyseur est conçu comme un système modulaire dont l'architecture générale est la suivante :



Pour faciliter la présentation des différents modules, nous les structurons en parties logiques dont chacune est référencée par le nom du module suivi d'un entier. Ces noms sont repris dans les Listings.

9.1.1. NATLNG

C'est le module directeur. Il appelle les autres modules et contient la structure syntaxique.

NATLNG 1.- Le module directeur commence par appeler le module DICTAB qui construit les tables dont les autres modules ont besoin.

NATLNG 2.- Un message est envoyé à l'utilisateur de l'analyseur syntaxique pour lui demander de formuler sa phrase :

FORMULEZ VOTRE PHRASE S.V.P.

NATLNG 3.- Appel à LEXIC
NATLNG 4.- Appel à SNTX1
NATLNG 5.- Appel à SNTX2
NATLNG 6.- Appel à SPIMP
Retour à NATLNG2

9.1.2. DICTAB

Ce module construit le dictionnaire et la table des catégories grammaticales. Les mots sont regroupés par longueur.

DICTAB1.- La construction du dictionnaire (annexe 1) est réalisée par une macro DICT (voir page suivante)

DICT &MOT, &N1, &N2, &N3, &LM

où &MOT - est une liste de mots $x_1, \ldots, x_m$ de même longueur &LM.

&N1 - est la liste des catégories grammaticales des x_i

&N2 - est la liste des numéros de chaque x_i à l'intérieur de sa catégorie grammaticale

&N3 - est la liste des traits syntaxiques et sémantiques liés aux x_i.

Exemple : DICT (être, peux, quel, sont, suis, vous), (10, 10, 30, 10, 10, 22), (2, 3, 1, 2, 2, 4), (01, 10, 00, 18, 10, 08), 4

signifie que :

être- est un verbe (10); le 2ème verbe (2); sous la forme infinitive

peux- est un verbe (10); le 3ème verbe (3); est au présent; ..

On constatera que "être", "sont" et "suis" ont tous la même composante N2, ici 2. Ce qui signifie qu'ils sont des formes différentes du même verbe qui est le 2ème de l'ensemble des verbes du dictionnaire.

La composante &N3 se présente comme suit :

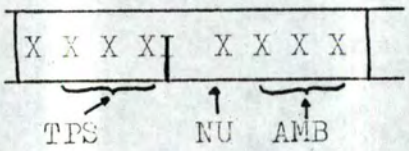
a) si $x_j \in V_{BD}$, &N3 indique la classe du nom correspondant :

CLASSE	&N3
AUTEUR	1
TITRE	2
COLLECTION	3

```
MACLIB=SOW.LNMLIB                                          USER-ID=FUN96     DATE=05/16/75


00000101 DICT     01010700:01012300 02/03/75 02/03/75

01010700          MACRO
01010800 &NOM     DICT  &MOT,&N1,&N2,&N3,&LM
01010900          LCLA  &A,&B
01011000 &B       SETA  N'&MOT
01011100 .LOOP    ANOP
01011200 &A       SETA  &A+1
01011300          AIF   (&A GT &B).FINDIC
01011400          AIF   (K'&MOT(&A) NE &LM).ERLONG
01011500          DC    C'&MOT(&A)'         LE MOT
01011600          DC    HL1'&N1(&A)'
01011700          DC    HL1'&N2(&A)'
01011800          DC    X'&N3(&A)'
01011900          AGO   .LOOP
01012000 .ERLONG  MNOTE *,'&MOT(&A) ERR LONG'
01012100          AGO   .LOOP
01012200 .FINDIC  ANOP
01012300          MEND



00000106 ENTD     01018000:01018700 02/03/75 02/03/75

01018000          MACRO
01018100 &NOM     ENTD  &LM,&NM
01018200          DS    0F
01018300 DIC&LM   DS    0CL6                1ERE  L-ENTREE
01018400          DC    F'0'
01018500          DC    HL1'0'              LIEN VERS 2EME L-ENTREE
01018600          DC    HL1'&NM'            NOMBRE DE L-MOTS
01018700          MEND



00000102 MAPMO    01012400:01013700 02/03/75 02/03/75

01012400          MACRO
01012500 &NOM     MAPMO &LONGMO
01012600          LCLA  &A,&B,&LM
01012700 &B       SETA  N'&LONGMO
01012800 .LOOP    ANOP
01012900 &A       SETA  &A+1
01013000          AIF   (&A GT &B).FINLM
01013100 &LM      SETA  &LONGMO(&A)         &LM CONTIENT LONG-MOT
01013200          L     5,=A(DIC&LM)        METTRE DANS 5 AD-L-ENTREE
01013300 &LM      SETA  &LM-1
01013400 &LM      SETA  &LM*4               &LM  POINTE VERS FW DE PMOT
01013500          ST    5,PMOT+&LM          GARNIR PMOT
01013600          AGO   .LOOP
01013700 .FINLM   MEND
```

EDITEUR	4
ANNEE	5
COTE	6
FORMAT	7

b) si $x_j \in V_G$, &N3 indique selon le cas le temps (TPS) le nombre (NU) : singulier, pluriel, l'ambigüité possible (AMB)



TPS = { 0 - infinitif
1 - présent

NU = { 0 - singulier
1 - pluriel

AMB = { 0 - nom ambigu
1 - verbe - nom
2 - adjectif - nom
3 - article - pronom

Une macro :

ENDT &K, &NML

met en place l'entrée DIC &K des mots de longueur &K et indique dans une zone de cette entrée le nombre &NML de mots qui y figurent.

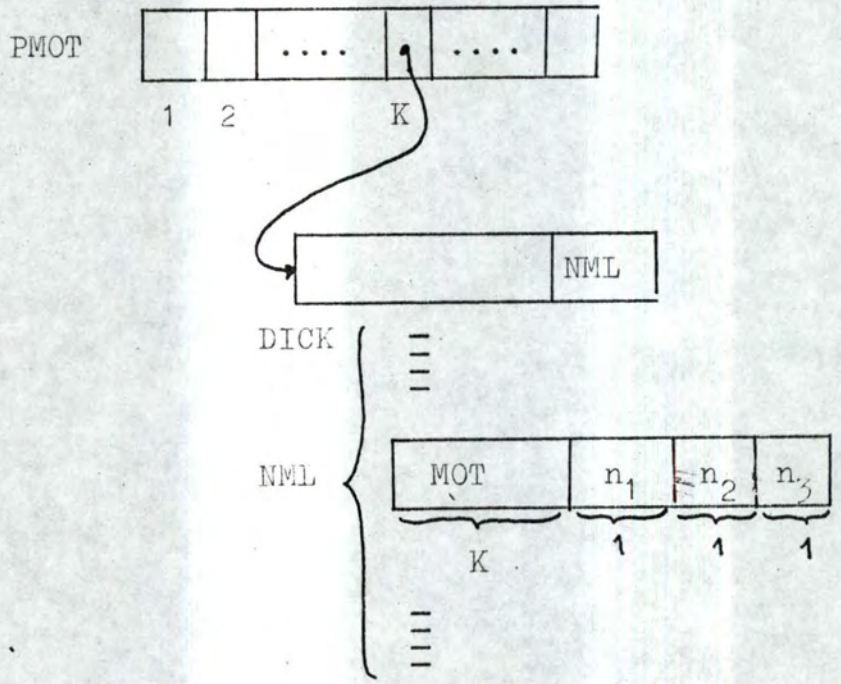
Ex. : ENDT 1, 7 met en place l'entrée DIC 1

des mots de longueur 1 qui sont au nombre de 7.

Enfin, une macro

MAPMO &LONGMOT

garnit une zone de pointeurs PMOT vers les différentes entrées DIC&K du dictionnaire



L'entrée d'un mot de longueur K est de K + 3 octets

DICTAB2 - construction de la table TABGR des catégories grammaticales avec la macro MATGR (voir page suivante)

MATGR &CAT, &K, &VOIS, &NT

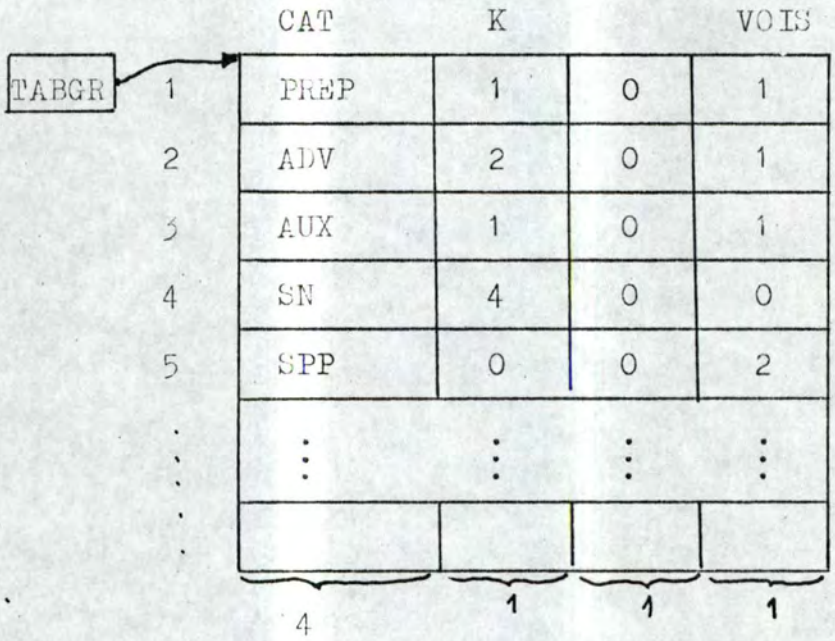
CAT - est le nom de la catégorie
K - sa valeur de sortie
VOIS - l'indicateur qui prend les valeurs suivantes :

1 si IND = /
0 si IND = |
2 si IND = \

```
MACLIB=SOW.LNMLIB                                                    USER-ID=FUN96     DATE=05/16/75


00000104 MATGR     01015700:01017100 02/03/75 02/03/75

01015700           MACRO
01015800 &NOM      MATGR &CAT,&K,&RD,&VOIS,&NT
01015900           LCLA  &A,&B,&C
01016000 &B        SETA  N'&CAT
01016100 &C        SETA  &NT*35
01016200           ORG   TABGR+&C
01016300 .LOOP     ANOP
01016400 &A        SETA  &A+1
01016500           AIF   (&A GT &B).FINTG
01016600           DC    CL4'&CAT(&A)'       C-GRAM,POIDS, RED-DIR,VOIS
01016700           DC    HL1'&K(&A)'
01016800           DC    HL1'&RD(&A)'
01016900           DC    HL1'&VOIS(&A)'
01017000           AGO   .LOOP
01017100 .FINTG    MEND



00000107 MSCAN     01018800:01021200 02/03/75 02/03/75

01018800           MACRO
01018900 &NOM      MSCAN &BASE,&DIR,&K,&I,&DEP,&L
01019000           L     5,A&BASE
01019100           L     6,A&DIR
01019200           BCTR  6,0
01019300           LR    8,&K
01019400           MH    8,K20
01019500           AR    6,8
01019600           USING &BASE,5
01019700           USING &DIR,6
01019800           XR    8,8
01019900           AR    6,&I                INCREMENT VECT-DIRECT
01020000           IC    8,&DIR
01020100           LTR   8,8
01020200           BC    8,ITERKV
01020300           MH    8,K7
01020400           AR    5,8                 5 POINTE DANS TABGR
01020500           AIF   (&L GT 1).MOT4      SI  MOT A RETOURNER
01020600           XR    8,8
01020700           IC    8,&BASE+&DEP        8 CONTIENT OCTET DE BASE
01020800           AGO   .FINSC
01020900 .MOT4     MVC   ZCAT(4),&BASE+&DEP  ZCAT CONTIENT MOT
01021000 .FINSC    DROP  5,6,7
01021100           STC   8,OCTET
01021200           MEND
```

Exemple : MATGR (PREP, ADV, AUX, SN, SPP),
(1, 2, 1, 4, 0), (0,0,0,0,0)
(1, 1, 1, 0, 2)

La valeur d'entrée des catégories est donnée implicitement par leur place par rapport au début de la table TABGR.
Il y a 34 catégories grammaticales ;



Chaque ligne de TABGR est de 7 octects dont le sixième n'est pas utilisé ici.

9.1.3. LEXIC

Ce module réalise l'analyse lexicale de la phrase pour dégager les différents mots, leur longueur, le nombre de mots.

Il reçoit comme paramètre la chaîne d'entrée ENTREE qui est la phrase, libellée par l'utilisateur.
Il utilise la macro MTRT qui génère des tables de TRT.

LEXIC1 - Lire le caractère suivant de ENTREE. Si le caractère lu est :

blanc - aller à LEXIC 3
" - aller à LEXIC 1
où ? - aller à LEXIC 2

LEXIC2 - On a lu le dernier mot. Le traiter et aller à la sortie de LEXIC.

LEXIC3 - On vient d'isoler un mot - LEXIC le place dans la chaîne PHRASE qui est un compactage de ENTREE.
La longueur du mot est mise dans le vecteur LONGMOT. Le contrôle est transféré à LEXIC1 après avoir fait plus 1 dans NM (nombre de mots).

LEXIC4 - On est en présence d'un nom de la base de donnée de la forme " ————". A la lecture des guillemets qui suivent, aller à LEXIC3.

Exemple :

ENTREE : LE LIVRE "AUTOMATED LANGUAGE PROCESSING" ....

PHRASE :

LE	LIVRE	AUTOMATED LANGUAGE PROCESSING	.

LONGMOT :

2	5	29	1

NM :

4

PHRASE, LONGMOT et NM sont situés dans NATLNG.

9.1.4. SNTX 1

C'est un programme de recherche simple dans le dictionnaire pour prendre les catégories grammaticales des mots de la phrase.

Il reçoit en entrée :

PNØT - liste des pointeurs vers les entrées du Dictionnaire PHRASE, LONGMOT, NM - qui viennent d'être construits par LEXIC.

SNTX11.- Lire dans PHRASE le mot suivant et sa longueur K dans LONGMOT. Si on a déjà lu NM mots, sortir de SNTX1

Chercher dans l'entrée DICK le mot lu. S'il est trouvé aller à SNTX 13.

SNTX12.- le mot n'est pas trouvé dans le dictionnaire. L'assimiler à nom commun et sortir le message "mot inconnu" Aller à SNTX 11.

SNTX13.- le mot est trouvé - Garnir trois lignes N1, N2, N3 avec respectivement les valeurs n1, n2, n3 qui figurent comme caractéristiques du mot dans le dictionnaire.
Aller à SNTX 11.

Exemple :

PHRASE :	LE	LIVRE	AUTOMATED	LANGUAGE	PROCESSING	.

N1	7	21	21	31
N2	2	4	0	1
N3	03	00	00	00

message :

AUTOMATED LANGUAGE PROCESSING mot inconnu

9.1.4. SNTX 2

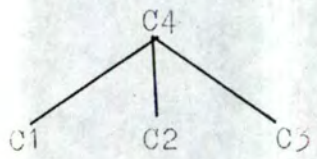
Ce module construit la structure partielle de la phrase.

L'arbre syntaxique est représenté dans une matrice STRUC (5, 20) où les lignes représentent les niveaux de l'arbre, les colonnes les rangs des éléments :

$$STRUC(i, j) \neq 0 \qquad i = 1,2,3,4,5 \qquad 0 < j < 20$$

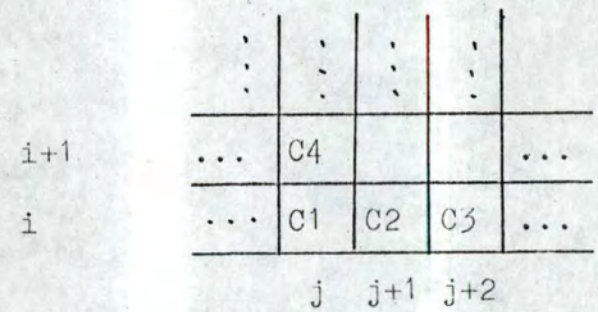
indique la présence d'un noeud de rang j et de niveau i.

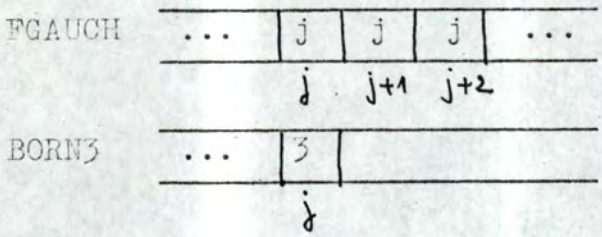
Supposons par exemple trois catégories grammaticales C1, C2, C3 qui forment un intervalle



de numérateur C4 et telles que dans STRUC, C1, C2, C3 sont de niveau i et de rangs respectifs, j, j + 1, j + 2.

Alors C4 est mis au niveau i + 1 dans la colonne j (relation père-fils le plus à gauche)





Une ligne de pointeurs FGAUCH indique pour chaque intervalle, l'élément le plus à gauche (relation frère-frère aîné) - La ligne BORN3 indique le nombre d'éléments de l'intervalle (nombre de fils du père).

SNTX2 reçoit en entrée la table des catégories grammaticales TABGR et construit la matrice STRUC dans NATLNG à partir de la ligne N1 des catégories grammaticales.

SNTX21 - Déterminer dans N1 le prochain intervalle en consultant dans la table TABGR les indicateurs des catégories de N1. Cette consultation est faite par la macro :

MSCAN &BASE &DIR &K &I &DEP

qui recherche dans une table &BASE les lignes référencées par un vecteur de pointeurs &DIR et dans chacune de ces lignes l'élément de rang &DEP - &K - &I sont des indices qui permettent de localiser les éléments de &DIR.
Ainsi l'appel

MSCAN TABGR STRUC RK RI

permet de prendre dans TABGR la valeur contenue dans le 7è octet (qui est l'indicateur de la catégorie grammaticale) correspondant au noeud de STRUC dont le numéro de ligne et de colonne sont contenus dans RK et RI.

N1 est parcourue de la gauche vers la droite. Si tous les intervalles sont traités, aller à SNTX23.

SNTX22 - faire la somme des valeurs de sortie des catégories de l'intervalle et mettre la valeur obtenue à la place du numérateur dans STRUC. Aller à SNTX21.

SNTX23 - appeler SNTX3 qui va compléter la structure syntaxique.

SNTX24 - imprimer la structure syntaxique - sortie de SNTX2.

9.1.5. SNTX 3

Ce module est l'implémentation du réseau grammatical élargi (A.G.). Son but est d'achever de construire la structure de la phrase et de dégager le type de la phrase ainsi que les fonctions grammaticales.

Les actions SETR 1, SETR 2, TEST 1 sont réalisées par des macros de même nom. Les actions MOUV, CONS, NEXT, TO, sont simplement réalisées par des instructions Assembleur . La reconnaissance des catégories CAT est faite par un TRT. La logique du réseau est présentée en 6.3. SNTX3 se déroule de la façon suivante :

SNTX30 - lire la catégorie grammaticale (numérateurs) de la structure partielle. Soit CAT la catégorie lue. Aller à TRCAT.

SNTX32 - TRSPP - traitement du syntagme prépositionnel qui est soit un complément nominal, soit un complément verbal, soit un complément indéterminé.

MACLIB=SOW.LNMLIB USER-ID=FUN96 · DATE=05/16/75

```
00000111 SETR1    01033100:01034300 02/28/75 02/28/75

01033100          MACRO
01033200 &NOM     SETR1 &CAT
01033300          XR    7,7
01033400          L     6,=A(KREL)
01033500          IC    7,0(6)
01033600          L     8,=A(Z&CAT)
01033700          BCTR  8,0
01033800          L     9,=A(K&CAT)
01033900          XR    10,10
01034000          IC    10,0(9,7)
01034100          AR    8,10
01034200          STC   RI,0(RK,8)
01034300          MEND


00000109 TEST1    01038300:01039000 03/04/75 03/04/75

01038300          MACRO
01038400 &NOM     TEST1 &CAT,&BR
01038500          XR    8,8
01038600          L     9,=A(Z&CAT)
01038700          LH    8,0(9,RK)
01038800          LTR   8,8
01038900          BC    8,&BR
01039000          MEND



00000112 SETR2    01036700:01038200 03/04/75 03/04/75

01036700          MACRO
01036800 &NOM     SETR2 &CAT,&DEP
01036900          XR    7,7
01037000          L     6,=A(KREL)
01037100          IC    7,0(6)
01037200          XR    8,8
01037300          XR    6,6
01037400          L     9,=A(KN)
01037500          IC    8,0(9,7)
01037600          S     8,=A(&DEP+1)
01037700          L     5,=A(ZSN)
01037800          LA    5,0(5,RK)
01037900          IC    6,0(5,8)
01038000          L     5,=A(Z&CAT)
01038100          STC   6,0(RK,5)
01038200          MEND
```

SNTX33 - TRVER - traitement du syntagme verbal, positionne en plus le sujet et traite les duplications s'il y a une relative.

SNTX34 - TRQUE - indiquer la présence d'une relative introduite par "que".

SNTX35 - TRQUI - indique la présence d'une relative introduite par "qui".

SNTX36 - TRNEG - traitement de la négation

SNTX37 - TRUNION - traitement du trait d'union comme indicateur d'inversion et d'une forme interrogative.

SNTX38 - TRQ1 - traitement de la catégorie Q1

SNTX39 - TRQ2 - traitement de la catégorie Q2

SNTX310 - TRCONN - traitement des connecteurs qui n'est pas implémenté ici.

SNTX311 - TRQ3 - traitement de Q3 - signes d'interrogation.

9.1.6. SPIMP

C'est un module d'impression qui permet de sortir les fonctions grammaticales et le type de la phrase tels qu'ils sont construits dans SNTX3.

9.2 Quelques caractéristiques numériques

L'analyseur est implémenté sur Siemens. L'ensemble des modules ne dépasse pas 1800 lignes assembleurs sources. Le segment objet occupe moins de 2 K (1777 octets mémoire) L'analyseur traite 15 phrases de longueurs variables (mais ne dépassant pas 20 mots) en 3,55 de temps CPU.

A.1. DICTIONNAIRE

a/VER/PRES/SG
à/PREP
l/ART/SG
m/PRO/SG
n/NEG
ai/VER/PRES/SG
ce/N/SG
de/PREP
du/PREP
en/PREP
et/CONN
il/PRO/SG
je/PRO/SG
la/ART/SG
le/ART/SG
ne/NEG
ou/CONN
un/ART
AN1/INT/ANNEE
AN2/INT/ANNEE
des/PREP/PL
est/VER/PRES/SG
les/ART/PL
par/PREP
pas/NEG
que/RQUE
qui/RQUI
son/ADJ1/SG
sur/PREP
une/ART/SG
vos/ADJ1/PL
AUT1/NPR/AUTEUR
AUT2/NPR/AUTEUR
avez/VER/PRES/PL
chez/PREP
dont/RDONT
COL1/NPR/COLLECTION
COL2/NPR/COLLECTION
cote/N/SG
dans/PREP
elle/PRO/SG
être/VER/INF
peux/VER/PRES/SG
quel/Q3/SG
sont/VER/PRES/PL
suis/VER/PRES/SG
vous/PRO/PL
année/N/SG
avant/PREP
avoir/VER/INF
après/PREP
cotes/N/PL
depuis/PREP
EDIT1/NPR/EDITEUR
EDIT2/NPR/EDITEUR
libre/ADJ2/SG
liste/N/SG
livre/N/SG

quels/Q3/PL
titre/N/SG
revue/N/SG
TITR1/ADJ2/TITRE
TITR2/ADJ2/TITRE
années/N/PL
auteur/N/SG
donner/VER/INF
donnez/VER/PRES/PL
écrire/VER/INF
libres/ADJ2/PL
livres/N/PL
numéro/N/SG
pouvez/VER/PRES/PL
publier/VER/INF
quelle/Q3
relatif/ADJ2/SG
éditeurs/N/PL
éditions/N/PL
journaux/N/PL
ouvrages/N/PL
posséder/VER/INF
emprunter/VER/INF
fascicule/N/SG
collection/N/SG
disponible/ADJ2/SG
fascicules/N/PL
collections/N/PL
disponibles/ADJ2/PL

A.2. CATEGORIES GRAMMATICALES

PREP	1	1	/	DET	16	0	/
ADV	2	0	/	*	17	0	
AUX	3	1	/	NEG	18	0	/
SN	4	4		NEGZ	19	0	
SPP	5	0		NPR	20	4	
COMP	6	0	/	N	21	4	
ART	7	0	/	PRO	22	4	
ADJ1	8	0	/	INT	23	4	
ADJ2	9	0		CONN	24	0	
VER	10	10		Q1	25	0	
SVER	11	10	/	Q2	26	0	
COP	12	10	/	RQUE	27	0	
MOD	13	0	/	RQUI	28	0	
SPRD	14	11		RQU3	29	0	
S	15	0		Q3	30	0	
				INV	31	0	
				FIN	32	0	

A.3. RESEAU GRAMMATICAL ELARGI

```
0. (S0)
1. (TRSN (CAT  SN)
         (TEST1 VER  SNT) (TEST1  INV  CODT)
         (SETR1  S  *) (SETR1 SUJ  *) (CONS  S) (NEXT  1)
    CODT (SETR1  COD  *) (NEXT  1)
     SNT (SETR1  SN  *) (CONS  SN) (NEXT 1))

2. (TRSPP (CAT  SPP)
          (TEST1  VER  CPNT) (TEST1  COD  CPVT)
     CPXT (SETR1  CPX  *) (NEXT 1)
     CPVT (SETR1  CPV  *) (NEXT 1)
     CPNT (SETR1  CPN  *) (NEXT 1))

3. (TRVER (CAT  SVER)
          (TEST1  SN  SPRDT)
     SUJ  (SETR2  SUJ  0) (CONS  S)
          (TEST1  RQUE  QUIT)
     QUET (SETR2  COD  0) (SETR1  RQUE  NIL) (TO  SPRDT)
     QUIT (TEST1  RQUI  SPRDT)
          (SETR2  SUJ  0) (SETR1  RQUI  NIL) (TO  SPRDT)
     SPRDT (SETR2  S  0) (SETR1  SPRD  *) (CONS  SPRD) (NEXT  1

4. (TRQUE (CAT  RQUE)
          (SETR1  RQUE  *) (NEXT 1))

5. (TRQUI (CAT  RQUI)
          (SETR1  RQUI  *) (NEXT 1))

6. (TR  INV (CAT  INV)
            (TEST1  SUJ  INVT) (SETR1  INV  NIL) (NEXT 2)
       INVT (MOUV  FORME "INT") (NEXT 1))
```

```
7. (TR  NEG (CAT  NEG)
            (MOUV  SIGNE "NEG") (NEXT 1))

8. (TR  Q2 (CAT  Q2)
           (MOUV  FORME "INT") (NEXT 1))

9. (TR  Q3 (CAT  Q3)
           (MOUV  FORME "INT") (SETR1  COD  *) (NEXT 1))
```

A.4. REGLES SEMANTIQUES

S-règles

S1.- FORME = dcl
SIGNE = pos
PRED = avoir ⟹ INFO (POSS (COD))
SUJ = vous
CLASS (COD.N) = TITRE
Ex. : vous avez le livre TITR1

S2.- FORME = int
SIGNE = pos
PRED = avoir ⟹ TEST (POSS (COD))
SUJ = vous
CLASS (COD.N) = TITRE
Ex. : avez-vous la revue TITR5 ?

S3.- FORME = dcl
SIGNE = pos ⟹ INFO (EQ (SUJ, COD))
PRED = être
CLASS (SUJ.N) = CLASS (COD.N)
Ex. : AUT1 est l'auteur de TITR1

S4.- FORME = int
SIGNE = pos ⟹ LIST (SUJ)
PRED = être
COD = qui, quel, quels, quelle, quelles
Ex. : Qui est l'auteur de TITR1 ?

S5.- FORME = dcl
SIGNE = pos
PRED.VER = être ⟹ INFO (DISPO(SUJ))
PRED.ADJ2 = disponible, libre
CLASS (SUJ.N) = TITRE
Ex. : le livre TITR1 est disponible

S6.- FORME = int
SIGNE = pos ⟹ TEST (REL)
PRED-2 = être
SUJ-2 = ce
Ex. : est-ce que le livre TITR1 est disponible ?

S7.- FORME = dcl
SIGNE = pos ⟹ LIST (COD)
PRED = donner, écrire
SUJ = NIL
Ex. : donnez les titres des articles de AUT1 que vous avez.

D-règles

D1.- SN.ART = un, une, quelques ⟹ (POUR PLUS X / ∇ ; △)
Ex. : un livre de AUT1

D2.- SN.ART = le, la, l, ⟹ (POUR LE X / ∇ ; △)
Ex. : la côte du livre TITR1

D3.- SN.ART = les ⟹ (POUR LES X / ∇ ; △)
Ex. : les articles de AUT1

D4.- SN.ADJ1 = tout, chaque ⟹ (POUR TOUT X / ▽ ; △)

Ex. : tout livre est de AUT1

N-règles

N1.- SN.N = livre, revue, article, ouvrage, titre ⟹ TITRE : ▽

Ex. : un livre

N2.- SN.N = auteur ⟹ AUTEUR : ▽

Ex. : l'auteur

N3.- SN.N = ANNEE ⟹ ANNEE : ▽

CPN.N = édition

Ex. : l'année d'édition

N4.- SN.N = côte ⟹ COTE : ▽

Ex. : la côte

N5.- SN.N = format ⟹ FORMAT : ▽

Ex. : un format

N6.- SN.N = livre, revue, article, ouvrage

CLASS (CPN.N) = AUTEUR ⟹ OUVR (CPN.N) : ▽

CPN.PREP = de, du, des

Ex. : l'article de AUT1.

N7.- SN.N = auteur

CLASS (CPN.N) = TITRE ⟹ AUT (CPN) : ▽

CPN.PREP = de, du, des

Ex. : l'auteur du livre TITR1.

N8.- SN.N = année
CLASS (CPN.1.N) = TITRE
CPN-1.PREP = de, du, des ⟹ AN (CPN-1) : ▽
CPN-2.N = édition
CPN-2.PREP = de
Ex. : l'année d'édition de TITR1

N9.- SN.N = côte
CLASS (CPN.N) = TITRE ⟹ COT (CPN) : ▽
CPN.PREP = de, du, des
Ex. : la côte de TITR1

N10.- SN.N. = format
CLASS (CPN.N) = TITRE ⟹ FORM (CPN) : ▽
CPN.PREP = de, du, des
Ex. : le format du livre TITR1

N11.- SN.N = collection ⟹ COLLECTION : ▽
Ex. : une collection

N12.- SN.N = collection
CLASS (CPN.N) = TITRE ⟹ COLL (CPN) : ▽
CPN.PREP = de, du, des
Ex. : la collection du livre TITR1

R-règles

R1.- SN-1.N = livre, revue, article, ouvrage
⟹ EQ (TITR(SN-1), SN-2)
CLASS (SN-2) = TITRE
Ex. : le livre "Automated Language Processing"

R2.- SN-1.N = collection
CLASS (SN-2) = COLLECTION ⟹ EQ (COLL (SN-1), SN-2)
Ex. : la collection "U"

R3.- SN-1.N = titre ⟹ (EQ(NAT(CPN-2.N), NAT (CPN-1.N)), EQ (AUT (CPN-2.N), CPN-1))
CLASS (CPN-1) = AUTEUR
CPN-1.PREP = de, du, des
CPN-2.N = livre, revue, article
CPN-2.PREP = de, du, des
Ex. : les titres des livres de AUT1

R4.- SN.N = livre, ouvrage ⟹ LIVRE (X)
Ex. : le livre TITR1

R5.- SN.N = revue ⟹ REVUE (X)
Ex. : la revue TITR2

R6.- SN.N = article ⟹ ARTICLE (X)
Ex. : un article

R7.- SN.N = journal ⟹ JOURNAL (X)
Ex. : le journal

R8.- SN.N = titre, livre, article, revues
PRED-VER = avoir
SUJ-1 = vous ⟹ POSS (COD)
PRED-2 ≠ NIL
Ex. : quels sont les articles de AUT1 que vous avez ?

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE
    000000                              1   NATLNG  START 0
                                        2           PRINT NOGEN
    00C00A                              3           ENTRY RETDIC,RETSNTX1,RETSNTX2
    CCE455                              4           ENTRY PHRASE,STRUC,LONGMOT,NM,PIV,BORN3,FGAUCH
    CCC28E                              5           ENTRY BORNC
    000174                              6           ENTRY N1,N2,N3
    0000A6                              7           ENTRY RETLEXIC
    CCCCDC                              8           ENTRY RETSPIMP
    00034C                              9           ENTRY ENTREE
    000002                             10           ENTRY KRET
    CCCCCD 05 CC                       11   DEBUT   BALR  12,0
    000002                             12           USING *,12
    000001                             13   RLM     EQU   1
    000002                             14   RNM     EQU   2
    00000B                             15   RI      EQU   11

                                       16   *  NATLNG1                          CALL DICTAB

    000002 07 00                       17   KRET    BCR   0,0
    000004 58 F0 C51E      000520      18           L     15,=V(DEBDICT)
    000008 07 FF                       19           BR    15
    00C00A 07 00                       20   RETDIC  BCR   0,0
    00C00C 07 00                       21   DEBTR   BCR   0,0
                                       22           RAZBL STRUC,0,200,0
                                       25           RAZBL N1,15,20C,C
                                       28           RAZBL ENTREE,40,256,0
                                       31           RAZBL PHRASE,40,200,0

                                       34   *  NATLNG2                          MESSAGE UTILISATEUR

** OBJECT TXT CARD # IS CCC1 **
    000036 92 C1 C2A4      00C2A6      35           MVI   ZIMP+4,X'C1'
    00C03A D2 1B C2A5C329  00C2A7 00032B 36           MVC   ZIMP+5(28),MESS
    CCC040 45 E0 C0E6      0000E8      37           BAL   14,IMPR
                                       38           RAZBL ZIMP,60,27,5
    00C04E 45 E0 C0E6      0000E8      41           BAL   14,IMPR
                                       42           RDATA ZONENT,TERMD
    00C08E 95 5C C34A      00C34C      62           CLI   ENTREE,C'*'
    000092 47 80 C0E2      0000E4      63           BE    TERMD
    000096 D2 83 C2A5C34A  00C2A7 00034C 64          MVC   ZIMP+5(132),ENTREE
    00C09C 45 E0 C0E6      0000E8      65           BAL   14,IMPR

                                       66   *  NATLNG3                          CALL LEXIC

    00C0A0 58 F0 C522      00C524      67           L     15,=V(DEBLEXIC)
    00C0A4 07 FF                       68           BR    15
    00C0A6 07 00                       69   RETLEXIC BCR  0,0

                                       70   *  NATLNG4                          CALL SNTX1

    00C0A8 58 F0 C526      00C528      71           L     15,=V(DEBSNTX1)
    0000AC 07 FF                       72   BR      BR    15
```

```
FLAG LOCTN OBJECT CODE     ADDR1  ADDR2  STMNT M SOURCE

    000DAE 07 00                          73   RETSNTX1 BCR   0,0
    000DB0 C7 00                          74   SUITT    BCR   0,0

                                          75   *        NATLNG5                          CALL SNTX2

    ** OBJECT TXT CARD # IS 0004 **
    000DB2 58 F0 C52A      000D2C         76            L     15,=V(DEBSNTX2)
    000DB6 07 FF                          77            BR    15
    000DB8 07 00                          78   RETSNTX2 BCR   0,0
    000DBA 92 45 C2A4      000DA6         79            MVI   ZIMP+4,X'45'
    000DBE D2 16 C2A5C532  000DA7 000D34  80            MVC   ZIMP+5(23),=C'FONCTIONS GRAMMATICALES    '
    000DC4 45 E0 C0E6      000EE8         81            BAL   14,IMPR
                                          82            RAZBL ZIMP,5C,23,5
    000DD2 45 E0 C0E6      000EE8         85            BAL   14,IMPR

                                          86   *    NATLNG6                              CALL SPIMP

    000DD6 58 F0 C52E      000D30         87            L     15,=V(DEBSPIMP)
    000DDA 07 FF                          88            BR    15
    000DDC 07 00                          89   RETSPIMP BCR   0,0
    000DDE 47 F0 C00A      000E0C         90            B     DEBTR
    000DE2 06 CC                          91   FINBQ1   BCTR  0,0
                                          92   TERMD    TERMD
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE.
    000000                              1   DICTAB  CSECT
                                        2           PRINT NOGEN
    000408                              3           ENTRY TABGR,PMOT,DEBDICT
    000000 05 C0                        4   DEBDICT BALR  12,0
    000002                              5           USING *,12


                                        6   *   DICTAB1                          CONSTRUCT. DICTIONNAIRE
                                        7           MAPMO (1,2,3,4,5,6,7,8,9,10,11)

    00005A 58 F0 C5FA       0005FC     30           L     15,=V(RETDIC)
    00005E 58 C0 C5FE       000600     31           L     12,=V(KRET)
    000062 07 FF                       32           BR    15
                                       33   TERMD   TERMD
** OBJECT TXT CARD # IS 0002 **
    000068                             36   PMOT    DS    0F
    000068 00                          37           DC    120HL1'0'
** OBJECT TXT CARD # IS 0003 **
** OBJECT TXT CARD # IS 0004 **
** OBJECT TXT CARD # IS 0005 **
    0000E0                             38   DICT    DS    0F
                                       39   ENT1    ENTD  1,10
                                       45   BGD1    DICT  (A,a,L,*,',',N),(10,1,7,24,18),                *
                                       46                 (1,1,1,3,1),(00,00,00,00,00),1
                                       58 1               *,',' ERR LONG
    0000F6 64180300                    63           DC    X'64180300'
    0000FA 4B1F0100                    64           DC    X'4B1F0100'
    0000FE 6F1F0200                    65           DC    X'6F1F0200'
    000102 60200101                    66           DC    X'60200101'
    000106 E3220600                    67           DC    X'E3220600'
    00010A C4                          68           DC    C'D'
    00010B 010204                      69           DC    X'010204'
                                       70   ENT2    ENTD  2,15
                                       76   BGD2    DICT  (AI,DF,DU,EN,ET,IL,JE,LE,LA,NE,OU),             *
                                       77                 (10,1,1,1,24,22,22,7,7,18,24),                *
                                       77                 (1,2,3,4,1,1,2,2,3,1,2),                      *
                                       77                 (10,04,00,00,00,00,00,03,03,00,00),2
                                      121          DICT  CE,21,10,00,2
    000152 E4D5                       126           DC    C'UN'
    000154 07                         127           DC    HL1'7'
    000155 04                         128           DC    HL1'4'
    000156 03                         129           DC    X'03'
    000157 E5C1                       130           DC    C'VA'
    000159 0A                         131           DC    HL1'10'
    00015A 0A                         132           DC    HL1'10'
    00015B 10                         133           DC    X'10'
    00015C D6D5                       134           DC    C'ON'
    00015E 16                         135           DC    HL1'22'
    00015F 05                         136           DC    HL1'5'
    000160 C0                         137           DC    X'00'
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE
                                        138   ENT3     ENTD  3,15
                                        144   BQD3     DICT  (AN1,AN2,DES,EST,LES,PAR,PAS,QUE,QUI),                *
                                        145                  (23,23,1,10,7,1,19,27,28),                             *
                                        145                  (1,2,5,2,4,6,1,1,1),                                   *
                                        145                  (F5,F5,0B,11,0B,00,00,00,00),3
                                        181                  (SON,SUR,UNE,VOS),                                     *
                                        182                  (8,1,7,8),(1,7,6,2),(02,00,00,08),3
0001B8 C4D6C9                           198                  DC    C'MOI'
0001BB 22                               199                  DC    HL1'34'
0001BC 05                               200                  DC    HL1'5'
0001BD 00                               201                  DC    X'00'
                                        202   ENT4     ENTD  4,18
                                        208   BQD4     DICT  (AUT1,AUT2,AVEZ,CHEZ,DONT,COL1,COL2,COTE,DANS,ELLE,   *
                                        209                  ETES),(20,20,10,1,29,9,9,21,1,22,10),                  *
                                        209                  (1,2,1,9,1,1,2,1,8,3,2),                               *
                                        209                  (F1,F1,03,00,00,F3,F3,00,00,00,18),4
                                        253                  DICT  (ETRE,PEUX,QUEL,SONT,SUIS,VOUS),                 *
                                        254                  (10,10,30,10,10,22),(2,3,1,2,2,4),                     *
                                        254                  (01,10,00,18,10,08),4
0001D0 E3D6E4E2                         278                  DC    C'TOUS'
0001E1 21                               279                  DC    HL1'33'
0001E2 01                               280                  DC    HL1'1'
0001E3 00                               281                  DC    X'00'
                                        282   ENT5     ENTD  5,20
                                        288   BQD5     DICT  (ANNEE,AVANT,AVOIR,APRES,COTES,DEPUIS,EDIT1,EDIT2),  *
                                        289                  (21,1,10,1,21,1,20,20),                                *
                                        289                  (2,10,1,11,1,12,3,4),                                  *
                                        289                  (00,00,01,00,08,00,F3,F3),5
                                        309 1                *,DEPUIS ERR LONG
                                        318                  DICT  (LIBRE,LISTE,LIVRE,QUELS,TITRE,REVUE,TITR1,TITR2),*
                                        319                  (9,21,21,30,21,21,9,9),                                *
                                        319                  (1,3,4,3,5,7,50,51),                                   *
                                        319                  (00,00,00,08,00,00,F2,F2),5
0001E4 D7C1D9D3C5                       351                  DC    C'PARLE'
0001E9 0A                               352                  DC    HL1'10'
0001EA 0B                               353                  DC    HL1'11'
0001EB 10                               354                  DC    X'10'
                                        355   ENT6     ENTD  6,20
                                        361   BQD6     DICT  (ANNEES,AUTEUR,DONNER,DONNEZ,ECRIRE,LIBRES),          *
                                        362                  (21,21,10,10,10,9),(2,6,4,4,5,1),                      *
                                        362                  (08,00,00,18,00,08),6
                                        386                  DICT  (LIVRES,NUMERO,POUVEZ,PUBLIER,QUELLE,REVUES,TITRES),*
                                        387                  (21,21,10,10,30,21,21),(4,11,3,6,2,7,5),               *
                                        387                  (08,00,18,00,00,08,08),6
                                        399 1                *,PUBLIER ERR LONG
                                        412   ENT7     ENTD  7,20
                                        418   BQD7     DICT  (AUTEURS,DERNIER,ECRIVEZ,EDITEUR,EDITION,JOURNAL),    *
                                        419                  (21,8,10,21,21,21),(6,3,5,8,9,10),                     *
                                        419                  (08,00,00,00,00,00),7
                                        443                  DICT  (NUMEROS,OUVRAGE,PREMIER,QUELLES,RELATIF),       *
                                        444                  (21,21,8,3,9),(11,12,4,4,2),                           *
                                        444                  (08,00,00,08,00),7
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

                                        464           DICT  (PARLENT,PUBLIES,TRAITER),(10,9,10),                *
                                        465                 (10,6,12),(18,08,00),7
 0003D2 C3D6D4C2C9C5D5                  477           DC    C'COMBIEN'
 0003D9 1A                              478           DC     HL1'26'
 0003DA 01                              479           DC     HL1'1'
 0003DB CC                              480   DC      DC     X'CC'
                                        481   ENT8    ENTD  8,10
                                        487   BQD8    DICT  (EDITEURS,EDITIONS,JOURNAUX,OUVRAGES,POSSEDER,       *
                                        488                 POSSEDEZ,RELATIFS),(21,21,21,21,10,10,9),           *
                                        488                 (8,9,10,12,7,7,2),(08,08,08,08,00,18,08),8
                                        516           DICT  (CONTIENT,TRAITENT),(10,10),(9,12),(10,18),8
                                        525   ENT9    ENTD  9,2
                                        531   BQD9    DICT  (EMPRUNTER,FASCICULE),(10,21),(8,13),(08,00),9
                                        540   ENT10   ENTD  10,4
                                        546   BQD10   DICT  (COLLECTION,DISPONIBLE,EST-CE-QUE,FASCICULES),       *
                                        547                 (21,9,25,21),(14,3,1,13),(00,02,00,08),10
                                        563   ENT11   ENTD  11,5
                                        569   BQD11   DICT  (COLLECTIONS,DISPONIBLES),(21,9),(14,3),(08,0A),11
                                        578           DICT  CONTIENNENT,10,9,18,11


                                        583   *   DICTAB2                          CINSTRUCT. TABLE CAT. GRAMM.

 0004D4 40404040                        584           DC    X'40404040'
 0004D8 040000                          585           DC    X'040000'
** OBJECT TXT CARD # IS 0028 **
 0004DB                                 586   TABGR   DS    0CL280
                                        587   TGR1    MATGR (PREP,ADV,AUX,SN,SPP),(1,2,1,4,0),(0,0,0,0,0),       *
                                        588                 (1,1,1,0,2),0
                                        609   TGR2    MATGR (COMP,ART,ADJ1,ADJ2,VER),(0,0,0,0,10),(0,0,0,0,0),   *
                                        610                 (2,1,1,2,0),1
                                        631   TGR3    MATGR (SVER,COF,MOD,SPRD,S),                               *
                                        632                 (10,10,0,11,15),(0,0,0,0,0),(1,1,1,2,3),2
                                        653   TGR4    MATGR (DET,*,NEG,NEGZ,NPR),(0,17,18,0,4),                  *
                                        654                 (0,0,0,0,0),(1,0,0,2,0),3
                                        675   TGR5    MATGR (N,PRO,INT,CONN,Q1),                                 *
                                        676                 (4,4,4,24,25),(0,0,0,0,0),(0,0,0,0,0),4
                                        697   TGR6    MATGR (Q2,RQUE,RQUI,RQU3,Q3),                              *
                                        698                 (26,27,28,29,30),(0,0,0,0,0),(0,0,0,0,0),5
                                        719   TGR7    MATGR (FIN,INV,QT,NIL),(31,32,33,34),(0,0,0,0),(0,0,0,0),6
** OBJECT TXT CARD # IS 0035 **
 0005DC 00000CE0                        737
 0005D4 00000110
 0005D8 00000164
 0005DC 000001C0
 0005E0 00000244
 0005E4 000002CC
 0005E8 00000340
 0005EC 000003DC
 0005F0 00000448
 0005F4 00000468
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE
    000000                              1   LEXIC   CSECT
                                        2           PRINT NOGEN
    000000                              3           ENTRY DEBLEXIC
    000000 C5 CC                        4   DEBLEXIC BALR  12,0
    000002                              5           USING *,12
    000001                              6   RENT    EQU   1
    000003                              7   RPH     EQU   3
    000004                              8   RLM     EQU   4
    000005                              9   RI      EQU   5
    000006                             10   RZ      EQU   6
    000007                             11   RLONG   EQU   7
    000009                             12   RBORNO  EQU   9
    00000A                             13   R1      EQU   10
    000002 94 0F CCAB      000CAD      14           NI    AIG+1,X'0F'
    000006 94 0F 00E1      000CE3      15           NI    AIG3+1,X'0F'
    00000A 58 70 C366      000368      16           L     RLONG,=V(LONGMOT)
    00000E 06 70                       17           BCTR  RLONG,0
    000010 58 90 C36A      000360      18           L     RBORNO,=V(BORNO)
    000014 06 90                       19           BCTR  RBORNO,0
    000016 58 30 C36E      000370      20           L     RPH,=V(PHRASE)
    00001A 58 10 C372      000374      21           L     RENT,=V(ENTREE)
    00001E 06 10                       22           BCTR  RENT,0
    000020 17 22                       23           XR    2,2
    000022 58 A0 C376      000378      24           L     R1,=A(1)
    000026 58 50 C37A      00037C      25           L     RI,=A(0)

                                       26   *   LEXIC1                          LIRE ENTREE

    00002A 07 00                       27   LOOP    BCR   0,0
    00002C 41 11 0001      000001      28           LA    RENT,1(RENT)
    000030 18 61                       29           LR    RZ,RENT
    000032 41 55 0001      000001      30           LA    RI,1(RI)
    ** OBJECT TXT CARD # IS 0001 **
    000036 59 50 C37E      000380      31           C     RI,=A(20)
    00003A 47 80 C152      000154      32           BE    ERR
    00003E DD 31 1000C15E          000160 33           TRT   0(50,RENT),TABL1
    000044 47 F2 C042      000044      34   TABR    B     TABR(2)
    000048 47 F0 C094      000096      35           B     MOT
    00004C 47 F0 C102      000104      36           B     QUOTE
    000050 47 F0 CC56      000058      37           B     FIN
    000054 47 F0 C0D2      0000D4      38           B     TRAITU
    000058 07 C0                       39   FIN     BCR   0,0
    00005A 96 F0 C0AB      000CAD      40           OI    AIG+1,X'F0'
    00005E 47 F0 C094      000096      41           B     MOT

                                       42   *   LEXIC2                          DERNIER MOT LU

    000062 07 00                       43   FINI    BCR   0,0
    000064 42 40 C067      000069      44           STC   RLM,MVC1+1
    000068 D2 09 30006000              45   MVC1    MVC   0(10,RPH),0(RZ)
    ** OBJECT TXT CARD # IS 0002 **
    00006E 58 80 C382      000384      46           L     8,=V(NM)
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

     000072 41 55 0001    000001          47          LA    RI,1(RI)
     000076 42 58 0000    000000          48          STC   RI,0(8)
     00007A 06 50                         49          BCTR  RI,0
     00007C 41 75 7000                    50           LA   RLONG,0(RI,RLONG)
     000080 92 01 7001                    51           MVI  1(RLONG),X'01'
     000084 42 A5 9001                    52          STC   R1,1(RI,RBORNO)

     000088 58 F0 C386    000388          53          L     15,=V(RETLEXIC)
     00008C 58 C0 C38A    00038C          54          L     12,=V(KRET)
     000090 07 FF                         55          BR    15
                                          56  TERMD   TERMD

                                          59  *  LEXIC3                          MOT VOCAB. GRAMM.

     000096 07 00                         60  MOT     BCR   0,0
     000098 18 41                         61          LR    RLM,RENT
     00009A 1B 46                         62          SR    RLM,RZ
     00009C 12 44                         63          LTR   RLM,RLM
     00009E 47 C0 C148    000140          64          BC    12,CCREC
     0000A2 07 00                         65  SUITM   BCR   0,0
     ** OBJECT TXT CARD # IS 0003 **
     0000A4 42 45 7000                    66          STC   RLM,0(RI,RLONG)
     0000A8 42 A5 9000                    67          STC   R1,0(RI,RBORNO)
     0000AC 47 00 C060    000062          68  AIG     BC    0,FINI
     0000B0 06 40                         69          BCTR  RLM,0
     0000B2 47 00 C060    000062          70  AIG3    BC    0,FINI
     0000B6 42 40 C0B9    0000BB          71          STC   RLM,MVC+1
     0000BA D2 09 30006000                72  MVC     MVC   0(10,RPH),0(RZ)
     0000C0 41 A4 A001                    73          LA    R1,1(RLM,R1)
     0000C4 41 34 3001                    74          LA    RPH,1(RLM,RPH)
     0000C8 47 00 C0DC    0000DE          75  AIG1    BC    0,TRUSUIT
     0000CC 47 00 C126    000128          76  AIG2    BC    0,TRQUOT
     0000D0 47 F0 C028    00002A          77          B     LOOP
     0000D4 07 00                         78  TRAITU  BCR   0,0
     0000D6 96 F0 C0C7    0000C9          79          OI    AIG1+1,X'F0'
     ** OBJECT TXT CARD # IS 0004 **
     0000DA 47 F0 C094    000096          80          B     MOT
     0000DE 07 00                         81  TRUSUIT BCR   0,0
     0000E0 94 0F C0C7    0000C9          82          NI    AIG1+1,X'0F'
     0000E4 42 A5 9001                    83          STC   R1,1(RI,RBORNO)
     0000E8 58 80 C376    000378          84          L     8,=A(1)
     0000EC 42 85 7001                    85          STC   8,1(RI,RLONG)
     0000F0 92 60 3000                    86          MVI   0(RPH),X'60'
     0000F4 41 33 0001    000001          87          LA    RPH,1(RPH)
     0000F8 41 55 0001    000001          88          LA    RI,1(RI)
     0000FC 41 AA 0001    000001          89          LA    R1,1(R1)
     000100 47 F0 C028    00002A          90          B     LOOP

                                          91  *  LEXIC4                          MOT VOC. BD

     000104 07 00                         92  QLOTE   BCR   0,0
     000106 41 11 0001    000001          93          LA    RENT,1(RENT)
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

     00010A 18 61                          94          LR    RZ,RENT
     00010C DD 31 1000C260         000262  95          TRT   0(50,RENT),TABL2
     ** OBJECT TXT CARD # IS 0005 **
     000112 17 88                          96          XR    8,8
     000114 43 81 0001     000001          97          IC    8,1(RENT)
     000118 59 80 C38E     000390          98          C     8,=A(64)
     00011C 47 70 C134     000136          99          BNE   QUOTFIN
     000120 96 F0 000B     000CCD         100          OI    AIG2+1,X'F0'
     000124 47 F0 C094     000C96         101          B     MOT
     000128 07 00                         102  TRQUOT  BCR   0,0
     00012A 94 0F 000B     000CCD         103          NI    AIG2+1,X'0F'
     00012E 41 11 0001     000001         104          LA    RENT,1(RENT)
     000132 47 F0 C028     000C2A         105          B     LOOP
     000136 07 00                         106  QUOTFIN BCR   0,0
     000138 96 F0 C0B1     000CB3         107          OI    AIG3+1,X'F0'
     00013C 47 F0 C094     000C96         108          B     MOT
     000140 07 00                         109  SORTIE  BCR   0,0
     000142 92 0C C362     000364         110          MVI   KQUOT,X'00'
     000146 47 F0 C028     000C2A         111          B     LOOP
     ** OBJECT TXT CARD # IS 0006 **
     00014A 07 00                         112  COREC   BCR   0,0
     00014C 58 40 C376     000378         113          L     R14M,=A(1)
     000150 47 F0 C0A0     000CA2         114          B     SUITM
     000154 07 00                         115  ERR     BCR   0,0
     000156 58 F0 C392     000394         116          L     15,=V(RETSPIMP)
     00015A 58 C0 C396     000398         117          L     12,=A(2)
     00015E 07 FF                         118          BR    15
     000160 00                            119  TABL1   DC    256HL1'0'
     ** OBJECT TXT CARD # IS 0007 **
     ** OBJECT TXT CARD # IS 0008 **
     ** OBJECT TXT CARD # IS 0009 **
     ** OBJECT TXT CARD # IS 0010 **
                                          120          MTRT  (64,125,127,75,111,96),(4,4,8,12,12,16),TABL1
     000262 00                            135  TABL2   DC    256HL1'0'
     ** OBJECT TXT CARD # IS 0018 **
     ** OBJECT TXT CARD # IS 0019 **
     ** OBJECT TXT CARD # IS 0020 **
     ** OBJECT TXT CARD # IS 0021 **
                                          136          MTRT  127,4,TABL2
     000364 00                            141  KQUOT   DC    HL1'0'
     ** OBJECT TXT CARD # IS 0024 **
     000368 00000000                      142
     00036C 00000000
     000370 00000000
     000374 00000000
     000378 00000001
     00037C 00000000
     000380 00000014
     000384 00000000
     000388 00000000
     00038C 00000000
     000390 00000040
```

```
FLAG LOCTN OBJECT CODE     ADDR1  ADDR2  STMNT # SOURCE

 000000                                      1  SNTX1    CSECT
 000000                                      2           ENTRY DEBSNTX1
 000000 05 C0                                3  DEBSNTX1 BALR  12,0
 000002                                      4           USING *,12
 000001                                      5  RLM      EQU   1
 000002                                      6  RNM      EQU   2
 000003                                      7  RDIC     EQU   3
 000004                                      8  RCT1     EQU   4
 000005                                      9  RBORNO   EQU   5
 000006                                     10  RPMOT    EQU   6
 000007                                     11  RPH      EQU   7
 000009                                     12  RN1      EQU   9
 00000A                                     13  RCT2     EQU   10
 00000B                                     14  RI       EQU   11
 000003                                     15  DEPDIC   EQU   3

 000002 07 00                               16  DEBLT    BCR   0,0
 000004 58 90 C1D6      000108              17           L     RN1,=V(N3)
 000008 58 70 C1DA      0001DC              18           L     RPH,=V(PHRASE)
 00000C 58 50 C1DE      0001E0              19           L     RBORNO,=V(LONGMOT)
 000010 58 20 C1E2      0001E4              20           L     RNM,=V(NM)
 000014 58 60 C1E6      0001E8              21           L     RPMOT,=V(PMOT)
 000018 17 44                               22           XR    RCT1,RCT1
 00001A 17 AA                               23           XR    RCT2,RCT2
 00001C 43 A2 0000      000000              24           IC    RCT2,0(RNM)
 000020 17 88                               25           XR    8,8

                                            26  *  SNTX11                          LIRE MOT

 000022 07 00                               27  BOUCL    BCR   0,0
 000024 17 11                               28           XR    RLM,RLM
 000026 43 15 0000      000000              29           IC    RLM,0(RBORNO)       RLM CONT LONG MOT I
 00002A 18 81                               30           LR    8,RLM
 00002C 06 80                               31           BCTR  8,0
 00002E 42 80 C04B      00004D              32           STC   8,COMP+1
 000032 42 80 C05D      00005F              33           STC   8,MVC+1
** OBJECT TXT CARD # IS 0001 **
 000036 4C 80 C138      00013A              34           MH    8,K4
 00003A 58 38 6000                          35           L     RDIC,0(8,RPMOT)     RPMOT POINTE VERS DICK
 00003E 43 43 0005      000005              36           IC    RCT1,5(RDIC)         RCT1 CONT NBR DE LMOTS
 000042 12 44                               37           LTR   RCT1,RCT1
 000044 47 C0 C124      000126              38           BC    12,ER1
 000048 41 33 0006      000006              39           LA    RDIC,6(RDIC)        RDIC POINTE VERS 1ER MOT
 00004C D5 00 3000 7000                     40  COMP     CLC   0(1,RDIC),0(RPH)    COMP MOT LU AVEC LMOT
 000052 47 80 008C      00008E              41           BE    TROUVE
 000056 41 31 3003                          42           LA    RDIC,DEPDIC(RLM,RDIC)
 00005A 46 40 C04A      00004C              43           BCT   RCT1,COMP           PASEER MOT DIC SUIVANT
 00005E D2 09 C14C 700C 00014E              44  MVC      MVC   ZIMP+5(10),0(RPH)

                                            45  *  SNTX12                          MOT INCONNU
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE
   000064 17 88                              46            XR    8,8
   000066 43 80 C13B      000139             47            IC    8,ZINC
   00006A 41 88 0001      000001             48            LA    8,1(8)
   ** OBJECT TXT CARD # IS 0002 **
   00006E 42 80 C13B      000139             49            STC   8,ZINC
   000072 92 41 C14B      00014D             50            MVI   ZIMP+4,X'41'
   000076 D2 0A C16FC13C  000171 00013E      51            MVC   ZIMP+40(11),MINC
   00007C 45 E0 C0CA      0000CC             52            BAL   14,IMPR
   000080 58 10 C1EA      0001EC             53            L     RLM,=A(0)
   000084 43 15 0000      000000             54            IC    RLM,0(RBORNO)
   000088 18 D9                              55            LR    13,RN1                SAVE VAL REG N1
   00008A 47 F0 C0AA      0000AC             56            B     LA2

                                             57    *       SNTX13                      GARNIR N1,N2,N3

   00008E 48 80 C136      000138             58    TROUVE  LH    8,K3
   000092 18 D9                              59            LR    13,RN1
   000094 41 31 3003                         60            LA    RDIC,DEFDIC(RLM,RDIC)
   000098 06 30                              61            BCTR  RDIC,0
   00009A D2 00 90003000                     62    MVCN1   MVC   0(1,RN1),0(RDIC)      GARNIR N1,N2,N2
   0000A0 4B 30 C134      000136             63            SH    RDIC,K1
   ** OBJECT TXT CARD # IS 0003 **
   0000A4 41 99 0014      000014             64            LA    RN1,20(RN1)
   0000A8 46 80 C098      00009A             65            BCT   8,MVCN1
   0000AC 41 9D 0001      000001             66    LA2     LA    RN1,1(13)             RETABLIR RN1 ET RN1+1
   0000B0 1A 71                              67            AR    RPH,RLM               RPH POINTE VERS MOT SUIV PHRASE
   0000B2 41 55 0001      000001             68            LA    RBORNO,1(RBORNO)
   0000B6 46 A0 C020      000022             69            BCT   RCT2,BOUCL



   0000BA 43 80 C13B      000139             70            IC    8,ZINC
   0000BE 58 F0 C1EE      0001F0             71            L     15,=V(RETSNTX1)
   0000C2 58 C0 C1F2      0001F4             72            L     12,=V(KRET)
   0000C6 07 FF                              73            BR    15

   0000C8 0A 1C                              74    TERMD   TERMD
   0000CA CA 16                              75 1  TERMD   SVC   28
                                             76 1          SVC   22
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE
    000000                              1   SNTX2    CSECT
                                        2            PRINT NOGEN
    000000                              3            ENTRY DEBSNTX2,RET32,RETSNTX3
    000000 05 C0                        4   DEBSNTX2 BALR  12,0
    000002                              5            USING *,12
    000002                              6   RI       EQU   2
    000003                              7   RJ       EQU   3
    000004                              8   RKV      EQU   4
    000009                              9   RSOM1    EQU   9
    00000A                             10   RSOM2    EQU   10
    00000B                             11   RK       EQU   11
    000005                             12   RTGR     EQU   5
    000006                             13   RSTRUC   EQU   6
    000014                             14   NMAX     EQU   20
    000007                             15   RIMP     EQU   7
    000002 07 00                       16   RET32    BCR   0,0
    000004 58 80 C386      000388      17            L     8,=V(TABGR)
    000008 58 90 C38A      00038C      18            L     9,=A(7)
    00000C 1B 89                       19            SR    8,9
    00000E 50 80 C2EE      000200      20            ST    8,ATABGR
    000012 58 80 C38E      000390      21            L     8,=V(STRUC)
    000016 58 90 C392      000394      22            L     9,=A(20)
    00001A 1B 89                       23            SR    8,9
    00001C 50 80 C2F2      000204      24            ST    8,ASTRUC
    000020 94 0F C159      00015E      25            NI    AIGBFG+1,X'0F'
    000024 58 B0 C396      000398      26            L     RK,=A(1)

                                       27   *   SNTX21                          DETERM. INTERVALLE
    000028 07 00                       28   NIVSUP   BCR   0,0
    00002A 17 22                       29            XR    RI,RI
    00002C 58 80 C2DE      0001EC      30            L     8,ANM
    000030 43 28 0000      000000      31            IC    RI,0(8)
    000034 17 AA                       32            XR    RSOM2,RSOM2          SOM2=0
    000036 07 00                       33   INTSUIV  BCR   0,0
    ** OBJECT TXT CARD # IS 0001 **
    00003A 58 40 C396      000398      34   SUIT31   L     RKV,=A(1)            KVOIS=1
    00003C 17 99                       35            XR    RSOM1,RSOM1          SOM1=0
    00003E 07 00                       36   CONSINT  BCR   0,0
                                       37            MSCAN TABGR,STRUC,RK,RI,4,1
    ** OBJECT TXT CARD # IS 0002 **
    000070 1A 98                       57            AR    RSOM1,8              SOM1=SOM1+K(I)
    000072 18 32                       58            LR    RJ,RI
                                       59   PRECJ    MSCAN TABGR,STRUC,RK,RJ,6,1
    0000A4 91 02 C2D3      0001D5      79            TM    OCTET,X'02'          PREC(J) ?
    ** OBJECT TXT CARD # IS 0003 **
    0000A8 47 80 C0B6      0000E8      80            BZ    BCTR                 NON
    0000AC 91 03 C2D3      0001D5      81            TM    OCTET,X'03'          OP LIAISON ?
    0000B0 47 10 C1A8      0001AA      82            BO    TRET
    0000B4 47 E0 C108      00010A      83            BNO   INTERV
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

     00C0B8 06 30                         84   BCTR     BCTR  RJ,0
     00C0BA 12 33                         85            LTR   RJ,RJ              J=0?
     00C0BC 47 C0 C120      00C122        86            BC    12,TRAIG           SI J NOT > 0 GOTO TRAIG
                                          87   SUIVJ    MSCAN TABGR,STRUC,RK,RJ,6,1
     00C0F0 91 01 C2D3      00C2D5       107            TM     OCTET,X'01'       MOT J A SUIV?
     00C0F4 47 10 C108      00C10A       108            BO    INTERV
     00C0F8 07 0C                        109   TESTV1   BCR   0,0
     00C0FA 91 01 C2DD      00C2DF       110   SUIT33   TM    ZVOIS,X'01'
     00C0FE 47 E0 C128      00C12A       111            BNO   SKSUIV
     00C102 92 00 C2DD      00C2DF       112   VZERO    MVI   ZVOIS,X'00'        VOIS = 0
     00C106 47 F0 C128      00C12A       113            B     SKSUIV
     00C10A 91 01 C2DD      00C2DF       114   INTERV   TM    ZVOIS,X'01'       EXISTE INTERVALLE?
     00C10E 47 10 C114      00C116       115            BO    ITERKV
     00C112 92 01 C2DD      00C2DF       116   VUN      MVI   ZVOIS,X'01'        VOIS = 1
     ** OBJECT TXT CARD # IS 0005 **
     00C116 41 44 0001      000001       117   ITERKV   LA     RKV,1(RKV)       KVOIS + 1
     00C11A 06 20                        118            BCTR  RI,0               I - 1
     00C11C 12 22                        119            LTR   RI,RI
     00C11E 47 20 C03C      00C03E       120            BC    2,CONSINT          SI I > 0 GOTO CONSINT
     00C122 96 F0 C183      00C185       121   TRAIG    OI    AIG+1,X'F0'        TRANSF AIG
     00C126 47 F0 C0F6      00C0F8       122            B     TESTV1


                                         123   *  SNTX22                          CONS. NUMERATEUR

     00C12A 07 00                        124   SKSLIV   BCR   0,0
     00C12C 18 32                        125   STRUC3I  LR    RJ,RI
     00C12E 06 30                        126            BCTR  RJ,0
     00C130 07 00                        127   SKI      BCR   0,0
     00C132 58 50 C2F2      00C2F4       128            L     5,ASTRUC
     00C136 41 80 0001      000001       129            LA    8,1(RK)
     00C13A 4C 80 C2AE      00C2B0       130            MH    8,K20
     00C13E 1A 58                        131            AR    5,8
     00C140 59 90 C39A      00C39C       132            C     RSOM1,=A(4)
     00C144 47 4C C14E      00C15C       133            BL    NONSIGN
     00C148 42 93 5000                   134            STC   RSOM1,0(RJ,5)      STRUCI = SOM1
     ** OBJECT TXT CARD # IS 0006 **
     00C14C 47 F0 C158      00C15A       135            B     AIGBFG
     00C150 07 00                        136   NONSIGN  BCR   0,0
     00C152 58 80 C39E      00C3A0       137            L     8,=A(17)
     00C156 42 83 5000                   138            STC   8,0(RJ,5)
     00C15A 47 00 C180      00C182       139   AIGBFG   BC    0,ADDSOM2
     00C15E 58 50 C2E2      00C2E4       140            L     5,ABORN3
     00C162 42 23 5000                   141            STC   RI,0(RJ,5)         BORN3(I,1) = I
     00C166 41 55 0014      000014       142            LA    5,20(5)
     00C16A 42 43 5001                   143            STC   RKV,1(RJ,5)        BORN3(I,2) = KVOIS
     00C16E 58 50 C2E6      00C2E8       144            L     5,AFGAUCH
     00C172 1A 53                        145            AR     5,RJ
     00C174 17 88                        146            XR    8,8
     00C176 42 28 5000                   147   LOOPFG   STC   RI,0(8,5)         FGAUCH(K) = I
     00C17A 41 88 0001      00C0C1       148            LA    8,1(8)           K = I,...,I+KVOIS
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

    000178 46 40 C174     000176          149           BCT   RKV,LOOPFG
    00017C 1A A9                          150  ADDSOM2  AR     RSOM2,RSOM1       SOM2 = SOM2 + SOM1
    ** OBJECT TXT CARD # IS 0007 **
    00017E 47 00 C18E     000190          151  AIG      BC    0,KSUIV
    000182 06 20                          152           BCTR  RI,0
    000184 12 22                          153           LTR   RI,RI
    000186 47 20 C034     000036          154           BC    2,INTSUIV         SI I > 0 GOTO INTSUIV


                                          155  *   SNTX23                          CALL SNTX3
    000190 07 00                          156  KSUIV    BCR   0,0
    000192 94 0F C183     000185          157           NI    AIG+1,X'0F'
    000196 96 F0 C159     000158          158           OI     AIGBFG+1,X'F0'
    00019A 58 F0 C3A2     0003A4          159           L     15,=V(DEBSNTX3)
    00019E 07 FF                          160           BR    15
    0001A0 07 00                          161  RETSNTX3 BCR   0,0
    0001A2 41 BB 0001     000001          162  ITERK    LA     RK,1(RK)
    0001A6 47 F0 C1BE     0001C0          163           B     FINARBR

    0001AA 1B A9                          164  TRET     SR    RSOM2,RSOM1
    0001AC 17 99                          165            XR    RSOM1,RSOM1
    0001AE 47 F0 C114     000116          166            B     ITERKV
    0001B2 D2 1F C30EC2B3 0003 10 0002B5  167  ROUTERS  MVC   ZIMP+20(32),ERSNTX1
    0001B8 45 E0 C242     000244          168           BAL   14,IMPR
    ** OBJECT TXT CARD # IS 0008 **
    0001BC 47 F0 C1C0     0001C2          169           B     SORTIE


    0001C0 07 00                          170  FINARBR  BCR   0,0
    0001C2 07 00                          171  SORTIE   BCR   0,0


                                          172  *   SNTX24                          IMPRES. ARBRE
    0001C4 92 45 C2FE     000300          173           MVI   ZIMP+4,X'45'
    0001C8 D2 13 C2FFC3B2 000301 0003B4   174           MVC   ZIMP+5(20),=C'STRUCTURE SYNTAXIQLE'
    0001CE 45 E0 C242     000244          175           BAL   14,IMPR
                                          176           RAZBL ZIMP,5C,20,5
    0001D2 45 E0 C242     000244          179           BAL   14,IMPR
    0001E0 07 00                          180  ARBRIMP  BCR   0,0
    0001E2 58 B0 C3A6     0003A8          181           L     RK,=A(5)
    0001E6 1A BB                          182  LOOPB    LR    8,RK
    0001E8 4C 80 C2AE     0002B0          183           MH    8,K20
    0001EC 58 60 C2F2     0002F4          184           L     RSTRUC,ASTRUC
    0001F0 58 70 C2F6     0002F8          185           L     RIMP,AZIMP
    ** OBJECT TXT CARD # IS 0009 **
    0001F4 1A 68                          186           AR    RSTRUC,8
    0001F6 17 AA                          187           XR    10,10
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE
     0001F8 58 80 C2DE    C002E0         188          L     8,ANM
     0001FC 43 A8 0000    000000         189          IC    10,0(8)
     000200 58 50 C2EE    0002F0         190   LOOPA  L     RTGR,ATABGR
     000204 17 88                        191          XR    8,8
     000206 43 86 0000    000000         192          IC    8,0(RSTRUC)
     00020A 4C 80 C2AC    0002AE         193          MH    8,K7
     00020E 1A 58                        194          AR    RTGR,8
     000210 D2 03 70005000               195          MVC   0(4,RIMP),0(RTGR)
     000216 41 66 0001    000001         196          LA    RSTRUC,1(RSTRUC)
     00021A 41 77 0006    000006         197          LA    RIMP,6(RIMP)
     00021F 46 A0 C1FE    0002000        198          BCT   10,LOOPA
     000222 45 E0 C242    000244         199          BAL   14,IMFR
     000226 46 B0 C1E4    0001E6         200          BCT   RK,LOOPB


     ** OBJECT TXT CARD # IS 0010 **
     00022A 58 F0 C3AA    0003AC         201          L     15,=V(RET23)
     00022E 58 C0 C3AE    0003B0         202          L     12,=V(DEB3)
     000232 07 FF                        203          BR    15


                                         204   TERMD  TERMD
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

    000000                              1     SNTX3    CSECT
                                        2              PRINT NOGEN
    000000                              3              ENTRY DEBSNTX3
    0005CC                              4              ENTRY RLIGN,KLIGN,KS
    000002                              5              ENTRY DEB3,RET23
    000000 05 C0                        6     DEBSNTX3 BALR  12,0
    000002                              7              USING *,12
    000001                              8     RSTRUC   EQU   1
    000003                              9     RI       EQU   3
    000004                             10     RZ       EQU   4
    000008                             11     RK       EQU   11
    000002 07 00                       12     DEB3     BCR   0,0
                                       13              RAZBL KREL,00,100,0
    000004 D2 08 C59EC8A6 0005A0 0008A8 16              MVC   PARAM(9),=C'AFFPOSDCL'
    000014 58 80 C826     000828       17              L     8,=V(NM)
    000018 43 58 0000     000000       18              IC    5,0(8)
    00001C 06 50                       19              BCTR  5,0
    00001E 42 50 C045     000047       20              STC   5,TRT+1
    000022 42 50 C031     000033       21              STC   5,MVC+1
    000026 42 50 C63E     000640       22              STC   5,ZNM
    00002A 17 BB                       23              XR    RK,RK
    00002C 17 22                       24              XR    2,2
    00002E 58 80 C82A     00082C       25              L     8,=V(STRUC)
    000032 D2 13 C6158014 000617       26     MVC      MVC   ZSTRUC(20),20(8)
    ** OBJECT TXT CARD # IS 0001 **
    000038 58 10 C82E     000830       27              L     RSTRUC,=A(ZSTRUC)
    00003C 58 40 C82E     000830       28              L     RZ,=A(ZSTRUC)



                                       29     *   SNTX30                          LIRE CAT. GRAMM.
    000040 07 00                       30     LOOP     BCR   0,0
    000042 58 E0 C832     000834       31              L     14,=A(RET)
    000046 DD FF 1000C63F        000641 32     TRT      TRT   0(256,RSTRUC),TBRCAT
    00004C 18 31                       33              LR    RI,RSTRUC
    00004E 1B 34                       34              SR    RI,RZ
    000050 41 33 0001     000001       35              LA    RI,1(RI)
    000054 43 80 C5E2     0005E4       36              IC    RK,ZK
    000058 47 F2 C056     000058       37     TABR     B     TABR(2)
    00005C 47 F0 C0EC     0000EE       38              B     TRSN
    000060 47 F0 C21E     000220       39              B     TRSPP
    000064 47 F0 C306     000308       40              B     TRVER
    000068 47 F0 C4F2     0004F4       41              B     TRNEG
    00006C 47 F0 C53A     00053C       42              B     TRQ2
    ** OBJECT TXT CARD # IS 0002 **
    000070 47 F0 C546     000548       43              B     TRQ3
    000074 47 F0 C492     000494       44              B     TRQUE
    000078 47 F0 C4BA     0004BC       45              B     TRQUI
    00007C 47 F0 C542     000544       46              B     TRCONN
    000080 47 F0 C090     000092       47              B     SORTIE
    000084 47 F0 C4FC     0004FE       48              B     TRUNION
    000088 07 00                       49     RET      BCR   0,0
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

     00008A 41 11 0001    000001          50          LA    RSTRUC,1(RSTRUC)
     00008E 47 F0 C03E    000040          51          B     LOOP
     000092 07 00                         52  SORTIE  BCR   0,0
     000094 58 F0 C836    000838          53          L     15,=V(RETSNTX3)
     000098 58 C0 C83A    00083C          54          L     12,=V(RET32)
     00009C 07 FF                         55          BR    15
     00009E 07 00                         56  RET23   BCR   0,0
     0000A0 92 45 C7A0    0007A2          57          MVI   ZIMP+4,X'45'
     ** OBJECT TXT CARD # IS 0003 **
     0000A4 D2 0D C7A1C8AF 0007A3 0008B1  58          MVC   ZIMP+5(14),=C'TYPE DE PHRASE  '
     0000AA 45 E0 C742    000744          59          BAL   14,IMPR
     0000AE 92 41 C7A0    0007A2          60          MVI   ZIMP+4,X'41'
                                          61          RAZBL ZIMP,5C,14,5
     0000BC 45 E0 C742    000744          64          BAL   14,IMPR
     0000C0 D2 04 C7A6C89F 0007A8 0008C1  65          MVC   ZIMP+10(5),=C'FORME'
     0000C6 D2 02 C7B0C5A4 0007B2 0005A6  66          MVC   ZIMP+20(3),FORM
     0000CC 45 E0 C742    000744          67          BAL   14,IMPR
     0000D0 D2 04 C7A6C8C4 0007A8 0008C6  68          MVC   ZIMP+10(5),=C'SIGNE'
     0000D6 D2 02 C7B0C5A1 0007B2 0005A3  69          MVC   ZIMP+20(3),SIGN
     ** OBJECT TXT CARD # IS 0004 **
     0000DC 45 E0 C742    000744          70          BAL   14,IMPR
     0000E0 58 F0 C83E    000840          71          L     15,=V(RETSNTX2)
     0000E4 58 C0 C842    000844          72          L     12,=V(KRET)
     0000E8 07 FF                         73          BR    15
                                          74  TERMD   TERMD

                                          77  *  SNTX31                          SYNTAGME NOMINAL
     0000EE 07 00                         78  TRSN    BCR   0,0
                                          79          MADD  KN,1
                                          89          SETR1 SN
                                         100          TEST1 VER,SNT
                                         106          TEST1 INV,CODT
     00014E 07 C0                        112  SUJINV  BCR   0,0
                                         113          MADD1 ZINV,-1
                                         119          SETR1 S
                                         130          MADD  KSUJ,1
                                         140          SETR1 SUJ
     0001BE 58 80 C82A    00082C         151          L     8,=V(STRUC)
     0001C2 41 83 803B                   152          LA    8,59(RI,8)
     0001C6 92 0F 8000                   153          MVI   0(8),X'0F'
     0001CA 47 F0 C20E    000210         154          B     SNT
     0001CE 07 00                        155  CODT    BCR   0,0
                                         156          MADD  KCOD,1
                                         166          SETR1 COD
     00020E 07 FE                        177          BR    14
     000210 07 00                        178  SNT     BCR   0,0
     000212 58 80 C82A    00082C         179          L     8,=V(STRUC)
     000216 41 83 8027                   180          LA    8,39(RI,8)
     00021A 92 04 8000                   181          MVI   0(8),X'04'
     00021E 07 FE                        182          BR    14

                                         183  *  SNTX32                          SYNTAGME PREPOSITIONNEL
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE
    000220 07 00                        184   TRSPP    BCR   0,0
                                        185            TEST1 VER,CPNT
                                        191            TEST1 COD,CPVT
    000242 07 00                        197   CPXT     BCR   0,0
                                        198            MADD  KCPX,1
                                        208            SETR1 CPX
    000282 07 FE                        219            BR    14
    000284 07 00                        220   CPVT     BCR   0,0
                                        221            MADD  KCPV,1
                                        231            SETR1 CPV
    0002C4 07 FE                        242            BR    14
    0002C6 07 00                        243   CPNT     BCR   0,0
                                        244            MADD  KCPN,1
                                        254            SETR1 CPN
    ** OBJECT TXT CARD # IS 0014 **
    000306 07 FE                        265            BR    14

                                        266   *  SNTX33                          SYNTAGME VERBAL

    000308 07 00                        267   TRVER    BCR   0,0
                                        268            MADD  KS,1
                                        278            MADD  KSPRD,1
                                        288            TEST1 SN,SPRDT
    000356 07 00                        294   SUJT     BCR   0,0
                                        295            MADD  KSUJ,1
                                        305            SETR2 SUJ,0
    0003A4 58 80 C82A     00082C        319            L     8,=V(STRUC)
    ** OBJECT TXT CARD # IS 0017 **
    0003A8 41 86 8038                   320            LA    8,59(6,8)
    0003AC 92 0F 8000                   321            MVI   0(8),X'0F'
    0003B0 95 00 C5AC     0005AE        322            CLI   KQUE,X'00'
    0003B4 47 70 C420     000422        323            BNE   QUET
    0003B8 95 00 C5AD     0005AF        324            CLI   KQUI,X'00'
    0003BC 47 70 C468     00046A        325            BNE   QUIT
    0003C0 07 00                        326   SENT     BCR   0,0
                                        327            SETR2 S,0
    0003F0 07 00                        341   SPRDT    BCR   0,0
    0003F2 07 00                        342   SUITV    BCR   0,0
                                        343            SETR1 SPRD
    000414 58 80 C82A     00082C        354            L     8,=V(STRUC)
    ** OBJECT TXT CARD # IS 0019 **
    000418 41 83 8027                   355            LA    8,39(RI,8)
    00041C 92 0B 8000                   356            MVI   0(8),X'0B'
    000420 07 FE                        357            BR    14
    000422 07 00                        358   QUET     BCR   0,0
                                        359            SETR3 COD,0
                                        369            MADD  KCOD,1
    000462 92 00 C5AC     0005AE        379            MVI   KQUE,X'00'
    000466 47 F0 C3BE     0003C0        380            B     SENT
    00046A 07 00                        381   QUIT     BCR   0,0
                                        382            SETR3 SUJ,0
    00048C 92 00 C5AD     0005AF        392            MVI   KQUI,X'00'
```

```
FLAG LOCTN OBJECT CODE    ADDR1  ADDR2  STMNT M SOURCE

    000490 47 F0 C3BE     0003C0          393           B     SENT

                                          394   *   SNTX34                        RELATIVE QUE
    000494 07 00                          395   TRQUE BCR   0,0
    000496 92 01 C5AC     0005AE          396         MVI   KQUE,X'01'
                                          397         MADD1 KREL,1
                                          403         MADD1 ZK,2
    00049A 07 FE                          409         BR    14

                                          410   *   SNTX35                        RELATIVE QUI
    ** OBJECT TXT CARD # IS 0022 **
    00049C 07 00                          411   TRQUI BCR   0,0
                                          412         TEST1 SN,TRQ3
    0004CE 92 01 C5AD     0005AF          418         MVI   KQUI,X'01'
                                          419         MADD1 KREL,1
                                          425         MADD1 ZK,2
    0004F2 07 FE                          431         BR    14

                                          432   *   SNTX36                        NEGATION
    ** OBJECT TXT CARD # IS 0023 **
    0004F4 07 00                          433   TRNEG BCR   0,0
    0004F6 D2 02 C5A1C8C9 0005A3 0008CB   434         MVC   SIGN(3),=C'NEG'
    0004FC 07 FE                          435         BR    14

                                          436   *   SNTX37                        TRAIT-D'UNION
    0004FE 07 00                          437   TRUNION BCR   0,0
                                          438         TEST1 SUJ,FINUNI
    000510 D2 02 C5A4C8CC 0005A6 0008CE   444         MVC   FORM(3),=C'INT'
    000516 07 00                          445   SUITUNI BCR   0,0
    000518 41 11 0002     000002          446         LA    RSTRUC,2(RSTRUC)
    00051C 47 F0 C03E     000040          447         B     LOOP
    000520 07 00                          448   FINUNI BCR   0,0
    000522 D2 02 C5A4C8CC 0005A6 0008CE   449         MVC   FORM(3),=C'INT'
                                          450         MADD1 ZINV,1
    000538 07 FE                          456         BR    14

                                          457   *   SNTX38                        CAT. GRAMM. Q1
                                          458   *   SNTX39                        CAT. GRAMM. Q2
                                          459   *   SNTX310                       CONNECTEURS
    00053A 07 00                          460   TRQ1  BCR   0,0
    00053C 07 00                          461   TRQ2  BCR   0,0
    00053E D2 02 C5A4C8CC 0005A6 0008CE   462         MVC   FORM(3),=C'INT'
    000544 07 00                          463   TRCONN BCR   0,0
    000546 07 FE                          464         BR    14

                                          465   *   SNTX311                       CAT. GRAMM. Q3
    000548 07 00                          466   TRQ3  BCR   0,0
                                          467         MADD  KCOD,1
                                          477         SETR1 COD
                                          488         MADD1 ZINV,1
    ** OBJECT TXT CARD # IS 0026 **
    000598 D2 02 C5A4C8CC 0005A6 0008CE   494         MVC   FORM(3),=C'INT'
```

PROGRAM: NATLNG
FILE: SOW.NATLNG.SO

SEGMENT: %ROOT

SEGMENT NUMBER: 1
START ADDR.: 000000
SEGMENT LENGTH: 002C69

NO. OF MODULES: 7
NO. OF ENTRY PTS.: 4C

NAME OF MODULE	LOAD ADDRESS	MODULE LENGTH	TRAITS	NO. OF ENTRYS	METHOD USED TO BIND MODULE
NATLNG	000C00	00054F		19	EXPLICIT
SPIMP	000550	0002CE		2	SOW.NATLNG.OB
SNTX2	000820	0003C8		4	SOW.NATLNG.OB
SNTX1	000BE8	000209		2	SOW.NATLNG.OB
LEXIC	000DF8	000398		2	SOW.NATLNG.OB
DICTAB	001190	000604		4	SOW.NATLNG.OB
SNTX3	001798	0008D1		7	SOW.NATLNG.OB

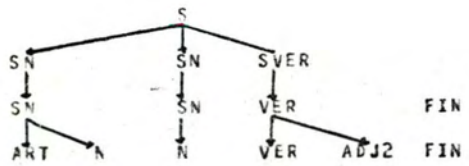
FORMULEZ VOTRE PHRASE S.V.P

LE LIVRE "FUNDAMENTAL ALGORITHMS" EST DISPONIBLE.

FUNDAMENTAL ALGORITHMS MOT INCONNU

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME DCL

SIGNE POS

FONCTIONS GRAMMATICALES

****

PRED EST DISPONIBLE

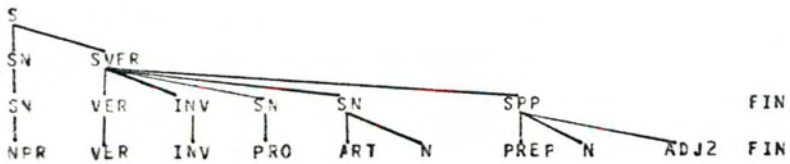
SUJ FUNDAMENTAL ALGORITHMS

FORMULEZ VOTRE PHRASE S.V.P

AUT2 EST-IL L'ALTEUR DL LIVRE TITR2?

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME INT

SIGNE POS

FONCTIONS GRAMMATICALES

****

PRED EST

SUJ AUT2

COD L AUTEUR

CPX DU LIVRTITR2

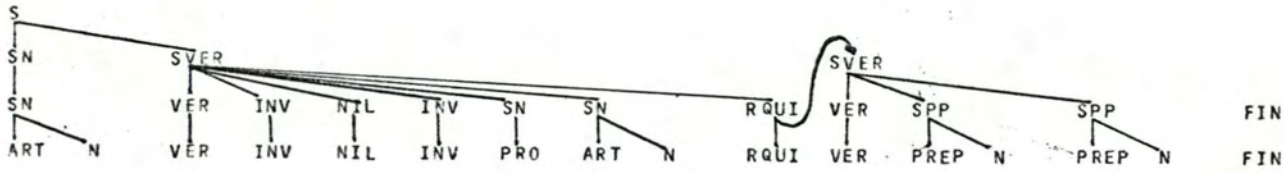
FORMULEZ VOTRE PHRASE S.V.P

LA SABENA A-T-ELLE UN VOL QUI VA DE BRUXELLES @ CONAKRY ?

SABENA	MOT INCONNU
VOL	MOT INCONNU
BRUXELLES	MOT INCONNU
CONAKRY	MOT INCONNU

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME INT

SIGNE POS

FONCTIONS GRAMMATICALES

****

PRED VA

CPV @ CONAKRY

CPV DE BRUXELLES

PRED A

SUJ LA SABENA

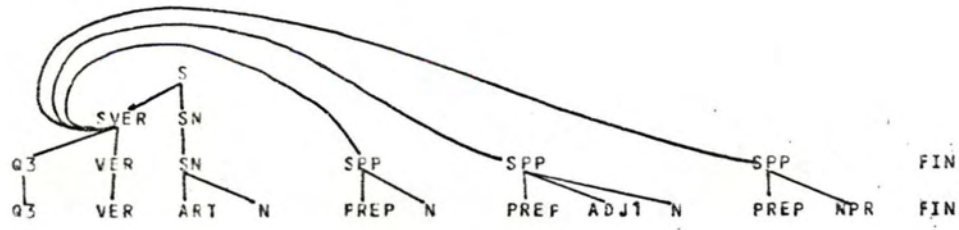
FORMULEZ VOTRE PHRASE S.V.P

QUELLE EST L'ANNEE DE PUBLICATION DU DERNIER LIVRE DE AUT1?

PUBLICATION MOT INCONNU

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME INT

SIGNE FOS

FONCTIONS GRAMMATICALES

****

PRED EST

SUJ L ANNEE

COD QUELLE

CPX DE AUT1

CPX DU DERNLIVRE

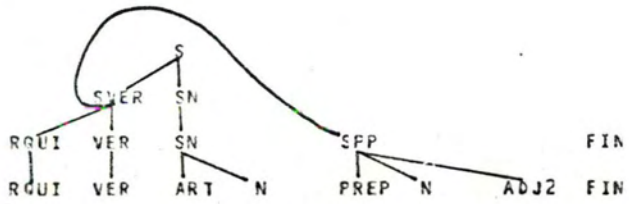
CPX DE PUBLICATION

FORMULEZ VOTRE PHRASE S.V.P

QUI EST L'AUTEUR DU LIVRE TITR1?

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME INT

SIGNE FOS

FONCTIONS GRAMMATICALES

****

PRED EST

SUJ L AUTEUR

COD QUI

CPX DU LIVRTITR1

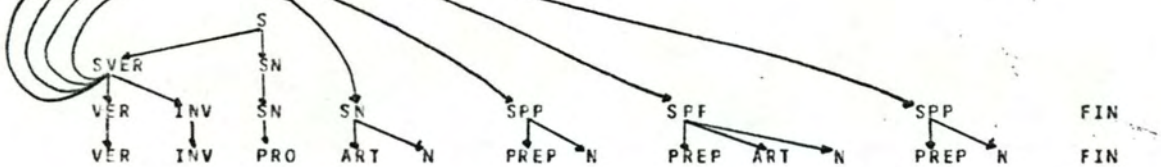
FORMULEZ VOTRE PHRASE S.V.P

AVEZ-VOUS LA THESE DE "W.A.WOODS" SUR LES SYSTEMES D'INTERROGATION?

THESE	MOT INCONNU
W.A.WOODS	MOT INCONNU
SYSTEMES	MOT INCONNU
INTERROGATION	MOT INCONNU

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME	INT
SIGNE	POS

FONCTIONS GRAMMATICALES

****

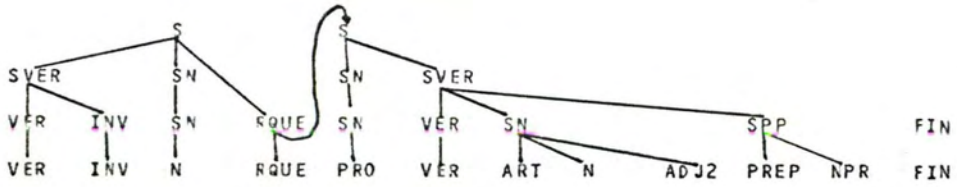
PRED	AVEZ
SUJ	VOUS
COD	LA THESE
CPX	D INTERROGATION

FORMULEZ VOTRE PHRASE S₍V₍P

EST-CE QUE VOUS AVEZ LE LIVRE TITR1 DE AUT2?

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME INT

SIGNE POS

FONCTIONS GRAMMATICALES

****

PRED AVEZ

SUJ VOUS

COD LE LIVRTITR1

COD CE

CPX DE AUT2

PRED EST

SUJ CE

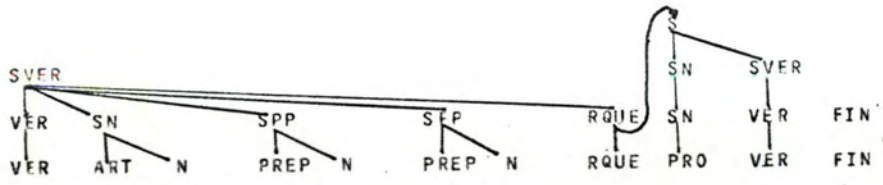
FORMULEZ VOTRE PHRASE S.V.P

DONNEZ LA LISTE DES OUVRAGES DE "CHOMSKY" QUE VOUS AVEZ.

CHOMSKY MOT INCONNU

STRUCTURE SYNTAXIQUE

****



TYPE DE PHRASE

****

FORME DCL

SIGNE POS

FONCTIONS GRAMMATICALES

****

PRED AVEZ

SUJ VOUS

COD LA LISTE

PRED DONNEZ

COD LA LISTE

CPX DE CHOMSKY

CPX DES OUVRAGES

BIBLIOGRAPHIE

[1]. BAR-HILLEL, Y. - A quasi arithmetical notation for syntactic description
Language, 29, nº 1, 1953, pp. 47-58

[2]. BAR-HILLEL, Y. - Syntactical and Semantical categories
Ed. Edwards P. The Ency. of Philosophy, V.8,
New York - Mac-Millan, 1967

[3]. BAR-HILLEL, Y. - Four lectures on Algebraic Linguistics and Machine Tranlation.
in Language and Information : Selected Essays on their theory and application.
Addison-Wesley, Redding - Mass. 1964

[4]. BAR-HILLEL, Y.- On categorial and Phrase structure grammars in op. cité.

[5]. BERTRAND, M.- Contribution à la synthèse automatique du Français.
Thèse de doctorat - Faculté des Sciences de Grenoble, 1964.

[6]. BOBROW, D.G.- Syntatic theories in computer Implementations
in Automated Language Processing. Ed. Borko H.
John Wiley, 1968.

[7]. BOBROW, FRASER, QUILLIAN - Automated Language Processing
in Annual Review of Information Science and Technology; vol. 2
Ed. Cuadra - John Wiley - 1967

[8]. CHOMSKY, N.- Aspects de la théorie syntaxique
L'ordre philosophique Seuil, 1971.

[9]. CHOMSKY, Miller - L'analyse formelle des langues naturelles
mathématiques et Sciences de l'homme.
Gauthier-Villars, 1968

[10]. CAUTIN, H.- Real English : A translator to enable
Language natural man-machine Conversation
Univ. of Pennsylvania, May 1969

[11]. DAVIS, R.M. - Programming Language Processors
in Avances in Computers, vol. 7
Academic Press, 1966.

[12]. EDMUNDSON, H.P. - Mathematical Models in Linguistics
and Language Processing.
in Automated Lang. Proc.
Ed. Harold B. John Wiley - 1968

[13]. GREEN, B.F. et d'autres - BASEBALL : an automatic
question answerer
in computers and thought
Ed. Frigenbaum and Feldman
McGraw Hill B.C., 1963

[14]. ICHBIAH, M. VALABREGUE, H.- L'analyse syntaxique auto-
matique de phrases françaises en vue de la
constitution de fichiers.
Congrès d'Informatique AFCET
Langages et communication graphique n° 3, 1970

[15]. JOYCE, F.- A computer model of transformational Grammar.
Math. Linguistics and Automatic
Language Processing, 9, Elsevier, 1971

[16]. KELLOG, C.H.- A natural language computer for on - line
data management.
Proc. AFIPS, 1968, Fall Joint Computer conf.,
vol. 33, pp. 473-492

[17]. KNUTH, D.E. - Fundamental Algorithms.
Computer Science and Information Proc.
Addison - Wesley, Publ. Co., 1969

[18]. LANGACKER, R.W. - French interrogatives : a transforma-
tional description.
Language, vol. 41, num. 4, 1965

[19]. MARCUS, S. - Algebraic Linguistics; Analytical Models.
Mathematics in Science and Engineering, vol. 29,
Academic Press, 1967

[20]. ROBERTS, R.- HELP. A question answering system.
Proc. AFIPS 1970 Fall Joint Computer Conference.

[21]. ROSENBAUM, P.S. - A grammar Base Question - Answering
Procedure
Com. ACM., 10, 10, oct. 1967

[22]. RUWET, N.- Introduction à la grammaire générative.
Recherches en Sciences Humaines
Plon, 1967

[23]. SAGER, N.- Syntatic Analysis of Natural Language.
Advances in computers, vol. 8
Academic Press, 1967

[24]. SALTON, G.- Automatic Phrase Matching
in. Readings in Automatic Language Processing
Ed. D.G. Hays - American Elsevier (1966)

[25]. SCHWARCZ, et d'autres - A deductive question answerer
for natural - Language inference
System Development Corporation
Santa Monica, nov. 1968

[26]. SIMMONS, R.F. - Automated Language Processing in .
Annual Review of information Science and
Technology.
Ed. Cuadra, vol. 1, 1966 - John Wiley.

[27]. SIMMONS, R.F. - Answering English Questions by computers
in Autom. Lang. Proc.
Ed. Harold Borko - John Wiley, 1968

[28]. SIMMONS, R.F. - Natural Language Question-Answering
Systems : 1969
Com. A.C.M., 13,1, Jan 1970, pp. 15-30

[29]. SIMMONS, SLOCUM, - Generating English Discourse from
Semantic Methods.
Com. A.C.M., 15, 10, oct. 1972, pp. 891-905.

[30]. SYRE, J.C. - LINNUS : Langage d'interrogation naturel utilisé dans le système SYNTEX.
Revue française d'Autom. Inform. Rech. Op. AFCET. DUNOD Sept. 1974

[31]. VEILLON, G.- Reconnaissance d'une figure dans une structure.
Congrès d'Informatique AFCET.
Langage et communication graphique, nº 3, 1970

[32]. WEIZENBAUM, J.- ELIZA : A computer Program For the Study of Natural Language Communication Between Man and Machine.
Com. A.C.M., 9, 1, Jan. 1966, pp. 36-45

[33]. WINOGRAD, T.- A program for understanding Natural Language
Cognitive Psychology, vol. 3, 1, 1972.

[34]. WOODS, W.A.- Semantics for a question-answering system.
Ph. D. thesis, Rep. NSF-19, Aiken Comp. Lab., Harvard Univ., Cambridge, Mass, 1967.

[35]. WOODS, W.A.- Procedural Semantics for a question-answering machine.
Proc. AFIPS 1968 - Fall Joint Comp. Conf., vol. 33, pp. 457-471
Ed. Thompson B. Co., Washington D.C.

[36]. WOODS, W.A. - Transition Network Grammars for Natural Language Analysis. Com. A.C.M., 13, 10, oct. 1970.

[37]. YNGVE, V.H. - A Framework for Syntatic Translation in Reading in Automatic Language Processing.
Ed. D.G. Hays - American Elsevier - 1966

ERRATA

Page	au lieu de	lire
iiii	considéré	considérée
6	pescent	percent
70	Backus-Nauer	Backus-Naur
88	sous-règles	S-règles
88	redoutantes	redondantes
100	s'occume	s'occupe
109	où ?	' . ' ou ' ? '
	"-aller à LEXIC1	"-aller à LEXIC4
115	Siemens	Siemens 4004-151